НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«На правах рукопису»
УДК 512.624.3

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“___” _____ 2020р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 113 «Прикладна математика»
(код і назва)

на тему: Системи лінійних заборон над скінченним полем

Виконав: студент 6 курсу, групи ФІ-83мн
(шифр групи)

Курінний Олег Вікторович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник доцент, кандидат фізико-математичних наук
Яковлєв С. В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2020 рік

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти: другий (магістерський) за освітньо–професійною програмою

Спеціальність: 113 «Прикладна математика»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
«___»_____20_ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Курінний Олег Вікторович
(прізвище, ім'я, по батькові)

1. Тема дисертації Системи лінійних заборон над скінченним полем, науковий керівник дисертації Яковлєв Сергій Володимирович, кандидат фізико-математичних наук, доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від _____р. № _____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження: інформаційні процеси в системах криптографічного зв'язку.
4. Предмет дослідження: системи лінійних заборон над скінченним полем.

5. Перелік завдань, які потрібно розробити:
1) сформулювати означення системи лінійних заборон та задачі її розв'язування; дослідити алгебраїчні та комбінаторні властивості системи лінійних заборон;
2) розглянути системи лінійних заборон із гарантовано однозначним розв'язком та дослідити питання відновлення розв'язку таких систем;
3) запропонувати алгоритми розв'язування часткових випадків систем лінійних заборон;
4) дослідити звідність задач розв'язування системи лінійних заборон до відомих теоретико-складнісних задач; встановити клас складності даної задачі.

6. Орієнтовний перелік ілюстративного матеріалу. Робота містить 2 таблиці та 1 рисунок.

7. Орієнтовний перелік публікацій. Результати роботи частково представлено у матеріалах двох міжнародних науково-практичних конференцій.

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Визначення орієнтовної теми дисертації та можливих напрямків дослідження по темі	вересень – жовтень 2018 року	виконано
2.	Опрацювання основної літератури на тему дослідження	листопад 2018 – вересень 2019	виконано
3.	Огляд наявних методів алгебраїчного криптоаналізу	жовтень 2019 – грудень 2019	виконано
4.	Дослідження складності задачі перевірки існування розв'язку систем лінійних заборон	січень 2020	виконано
5.	Дослідження алгебраїчних та комбінаторних властивостей систем лінійних заборон над скінченним полем	лютий 2020	виконано
6.	Дослідження систем лінійних заборон, що згенеровані фіксованим невідомим вектором; дослідження питання відновлення невідомого фіксованого вектору за системою лінійних заборон	березень 2020	виконано
7.	Розробка алгоритмів розв'язку деяких часткових випадків систем лінійних заборон; оцінка складності задач, пов'язаних з системами лінійних заборон	квітень 2020	виконано

Студент _____ О. В. Курінний
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ С. В. Яковлєв
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Кваліфікаційна робота містить: 86 стор., 1 рисунок, 2 таблиці, 35 джерел.

Метою роботи є розвиток та уточнення алгебраїчних моделей та методів криптоаналізу. Об'єктом дослідження є інформаційні процеси в системах криптографічного захисту. Предметом дослідження є система лінійних заборон та її властивості.

У даній роботі проведено огляд наявних методів алгебраїчного криптоаналізу та сформульовано задачу відновлення невідомого вектора за частковою інформацією, представленою у формі певних лінійних залежностей. Запропоновано формалізацію цієї задачі шляхом введення нотації системи лінійних заборон над скінченним полем. Побудовано критерій існування розв'язку систем лінійних заборон. Доведено ряд тверджень про кількість розв'язків системи лінійних заборон у випадку коли система лінійних заборон породжена фіксованим невідомим вектором. Отримано нетривіальну оцінку на точку насичення у випадку ненульових правих частин системи. Визначено задачі перевірки існування та пошуку розв'язку системи лінійних заборон, та доведено їх еквівалентність за Тюрінгом. Сформульовано ряд суміжних задач та доведено приналежність цих задач відповідним класам складності. Побудовано поліноміальні імовірнісні алгоритми перевірки існування та пошуку розв'язку для деяких часткових випадків. Також побудовано імовірнісний евристичний алгоритм пошуку декількох розв'язків системи лінійних заборон для деяких часткових випадків.

АЛГЕБРАЇЧНИЙ КРИПТОАНАЛІЗ, СИСТЕМИ ЛІНІЙНИХ ЗАБОРОН

ABSTRACT

Diploma work includes: 86 pages, 1 drawing, 2 tables, 35 references.

The goal of work is improving and clarifying models and methods of algebraic cryptanalysis. The research object are information processes in systems of cryptographic security. The research subject is system of linear restrictions and its properties.

In this work we overviewed existing methods of algebraic cryptanalysis and formulated problem of recovering unknown vector by partial information in the form of linear dependencies. We proposed formalization of this problem by introducing a notation of the system of linear restrictions over finite field. We constructed criterion of solution existence for the system of linear restrictions. We proved several claims about number of solutions for the system of linear restrictions generated by an unknown fixed vector. We get non-trivial upper bound on a saturation point for the system with non-zero right-hand side. We formulated decision and search problems for the system of linear restrictions and proved that these problems are Turing equivalent. We formulated several related problems and identified their complexity classes. We constructed polynomial probabilistic algorithms for decision and search problems in some partial cases. Also we constructed probabilistic heuristic algorithm for finding several solutions in some partial cases.

ALGEBRAIC CRYPTANALYSIS, SYSTEM OF LINEAR RESTRICTIONS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

B_n — множина булевих функцій від n змінних

$\langle f \rangle$ — ідеал функції f

$\mathrm{Ann}\,\langle f \rangle$ — анулятор ідеалу функції f

$\mathrm{poly}(x)$ — поліном від змінної x

$\leq_p$ — поліноміальна звідність

$\leq_T$ — звідність за Тюрінгом

$\mathbb{F}_{2^k}$ — поле Галуа, яке складається з 2^k елементів

$\mathbb{F}_{2^k}^*$ — мультиплікативна група поля $\mathbb{F}_{2^k}$

$\mathbb{F}_{2^k}^n$ — множина векторів довжини n над полем $\mathbb{F}_{2^k}$

(x, y) — скалярний добуток векторів x та y

$|A|$ — потужність множини A

XOR — додавання за модулем 2

0 — нульовий елемент поля $\mathbb{F}_{2^k}$

1 — одиничний елемент поля $\mathbb{F}_{2^k}$

$\overline{0}$ — вектор з нулів над полем $\mathbb{F}_{2^k}$

$\overline{1}$ — вектор з одиниць над полем $\mathbb{F}_{2^k}$

$\exists$ — квантор існування

$\forall$ — квантор загальності

$\langle a, a_0 \rangle$ — впорядкована пара, яка задає лінійну заборону $(a, x) \neq a_0$

$\mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_m)$ — діагональна матриця з елементами $\gamma_1, \gamma_2, \ldots, \gamma_m$

$x_{1:n}$ — перші n компонент вектору x

КНФ — кон'юнктивна нормальна форма

ВСТУП

Актуальність теми. Стандартні моделі алгебраїчного криптоаналізу розглядають залежність між відкритими текстами, шифротекстами та ключами у вигляді системи поліноміальних рівнянь над скінченним полем. Можна розглянути альтернативну задачу, в якій відомі лише обмеження на можливі значення деяких залежностей з невідомими параметрами. Дослідження такої задачі є доцільним, оскільки існує багато методів, що надають змогу отримати часткову інформацію про проміжні значення деяких параметрів в процесі шифрування. Ці методи можуть вказувати на те, що певні залежності з невідомими параметрами не можуть приймати деякий скінченний набір значень. Така інформація може бути отримана з побічного каналу зв'язку або з особливостей реалізації криптосистеми. З огляду на це, виникає задача відновлення невідомого вектора за частковою інформацією, представленою у формі певних лінійних залежностей.

Мета і завдання дослідження. Метою дослідження є розвиток та уточнення алгебраїчних моделей та методів криптоаналізу.

Для досягнення мети необхідно розв'язати такі завдання дослідження:

1) сформулювати означення системи лінійних заборон та задачі її розв'язування; дослідити алгебраїчні та комбінаторні властивості систем лінійних заборон;

2) розглянути системи лінійних заборон із гарантовано однозначним розв'язком та дослідити питання відновлення розв'язку таких систем;

3) запропонувати алгоритми розв'язування часткових випадків систем лінійних заборон;

4) дослідити звідність задачі розв'язування системи лінійних заборон до відомих теоретико-складнісних задач; встановити клас складності даної задачі.

Об'єктом дослідження є інформаційні процеси в системах криптографічного захисту.

Предметом дослідження є система лінійних заборон та її властивості.

Наукова новизна. В роботі отримані такі результати.

1) Визначено критерій існування розв'язку системи лінійних заборон. Побудовано поліноміальний імовірнісний алгоритм перевірки існування розв'язку для деяких часткових випадків. Доведено властивості розв'язків системи лінійних заборон у випадку нульових правих частин.

2) Сформульовано та доведено ряд тверджень про кількість розв'язків системи лінійних заборон у випадку коли система лінійних заборон породжена фіксованим невідомим вектором. Отримано нетривіальну оцінку на точку насичення у випадку ненульових правих частин системи.

3) Сформульовано задачі перевірки існування та пошуку розв'язку системи лінійних заборон. Доведено еквівалентність за Тюрінгом цих задач. Побудовано імовірнісний алгоритм пошуку розв'язку системи лінійних заборон у деяких часткових випадках. Побудовано імовірнісний евристичний алгоритм пошуку декількох розв'язків системи лінійних заборон у деяких часткових випадках. Сформульовано ряд суміжних задач та доведено приналежність цих задач відповідним класам складності.

Практичне значення. Результати даної роботи можуть бути використані як самостійний, так і допоміжний інструмент для алгебраїчного криптоаналізу потокових шифрів та криптосистем на лінійних кодах.

Апробації. Результати даної роботи були частково представлені на таких конференціях.

1) XVIII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (12-13 травня 2020 рік, м. Київ), назва

доповіді «Критерій розв'язку системи лінійних заборон над скінченним полем».

2) XII Міжнародна науково-практична конференція «Інтернет-Освіта-Наука 2020» (травень 2020 рік, м. Вінниця), назва доповіді «Складність задач, пов'язаних з системи лінійних заборон».

1 ОГЛЯД МЕТОДІВ АЛГЕБРАЇЧНОГО КРИПТОАНАЛІЗУ ТА ТЕОРІЇ СКЛАДНОСТІ

В даному розділі наведено огляд методів алгебраїчного криптоаналізу потокових шифрів. Основною задачею алгебраїчного криптоаналізу є побудова залежностей між відкритими текстами, шифротекстами та ключами у вигляді певної системи поліноміальних рівнянь над скінченним двійковим полем та розв'язок цієї системи. В загальному випадку така задача є обчислювально складною, але існують алгоритми розв'язку цієї задачі в певних часткових випадках. Після огляду таких алгоритмів описується ситуація, в якій про невідомий вектор можна отримати деяку часткову інформацію. Висувається ряд модельних припущень про вигляд такої інформації та формулюється задача відновлення невідомого вектора за частковою інформацією, представленої у формі лінійних залежностей. Також розглядаються означення та твердження з теорії складності, необхідні для подальших досліджень.

1.1 Методи алгебраїчного криптоаналізу потокових шифрів

Одним з видів симетричних криптосистем є *потокові шифри* [1]. Існує багато різних конструкцій потокових шифрів [2, 3]. У багатьох випадках потокові шифри мають ефективну реалізацію, тому вони часто використовуються на практиці в пристроях та системах телекомунікаційного зв'язку [4], де обов'язковою вимогою є висока швидкість шифрування та розшифрування. Поширеним типом потокових шифрів є *синхронний потоковий шифр* [5]. Цей шифр містить в собі такі складові.

1) Генератор гами: алгоритм, який приймає на вхід початковий стан

x_0 та породжує послідовність бітів $\gamma_1, \gamma_2, \ldots, \gamma_m$, яка називається *гамою*.

2) Алгоритм формування початкового стану x_0 за ключем та вектором ініціалізації.

3) Закон шифрування: найчастіше біт шифротексту обчислюється як XOR біту відкритого тексту з відповідним бітом гами.

Розповсюдженим класом атак на потокові шифри є *атаки на початковий стан* x_0. Припустимо в потоковому шифрі генератором гами є фільтрувальний генератор гами з регістром зсуву з лінійним зворотнім зв'язком [1], який має супроводжуючу матрицю C та функцію ускладнення f від n змінних з $\deg f = d$. В такому разі біти гами генеруються за формулою $\gamma_i = f(C^i \cdot x_0)$ для $i = 0, 1, \ldots, m$, де m — це довжина згенерованого відрізка гами.

За згенерованим відрізком гами можна скласти певну систему поліноміальних рівнянь степеня d. Зауважимо, що в загальному випадку це буде система рівнянь над полем $\mathbb{F}_{2^k}$, де $k \geq 1$, але оскільки будь-яку систему рівнянь над $\mathbb{F}_{2^k}$ можна представити у вигляді системи рівнянь над $\mathbb{F}_2$, то більшість наявних методів побудовано для $\mathbb{F}_2$. Задача полягає в тому, щоб за системою цих рівнянь відновити невідомий вектор x_0. Така задача в загальному випадку є $\mathcal{NP}$-повною при $d \geq 2$, але існують методи, які дозволяють оптимізувати або зменшити повний перебір:

1) побудова рівнянь-наслідків, які мають меншу за d степінь;

2) розв'язок системи рівнянь шляхом лінеаризації;

3) представлення системи рівнянь у вигляді булевої формули та застосування до неї SAT-розв'язувачів.

Розглянемо необхідні означення та твердження з теорії алгебраїчних многовидів [6]. Під B_n будемо позначати множину всіх булевих функцій від n змінних. *Ідеалом* булевої функції будемо називати множину $\langle f \rangle = \{g \cdot f, g \in B_n\}$. Якщо розглянути рівняння $f(x_1, x_2, \ldots, x_n) = 0$, то для всіх можливих наслідків цього рівняння типу $g(x_1, x_2, \ldots, x_n) = 0$ функції g будуть належати $\langle f \rangle$. Таким чином, ідеал функції f описує всі можливі рівняння-наслідки. *Анулятором* ідеалу функції f, позначається

$\mathrm{Ann}\,\langle f\rangle$, будемо називати множину $\{g \in B_n | g \cdot f = 0\}$. Для анулятора виконується така рівність: $\mathrm{Ann}\,\langle f\rangle = \langle f + 1\rangle$.

Нехай для $1 \leq i \leq m$ виконується $f(C^i \cdot x_0) = 1$. Припустимо, можна обрати такий $g \in \mathrm{Ann}\,\langle f\rangle$, що $\deg g < d$. Тоді $g(C^i \cdot x_0)$ буде рівнянням-наслідком, але вже меншого степеня. Переконаємось в цьому: якщо $g \in \mathrm{Ann}\,\langle f\rangle$, то $g(C^i \cdot x_0) \cdot f(C^i \cdot x_0) = 0$. Оскільки $f(C^i \cdot x_0) = 1$, то $g(C^i \cdot x_0) = 0$, отже отримали рівняння, степінь якого менший за d.

Розглянемо тепер випадок, коли для $1 \leq i \leq m$ виконується $f(C^i \cdot x_0) = 0$. Припустимо, що можна обрати $h \in \mathrm{Ann}\,\langle f + 1\rangle$ такий, що $\deg h < d$. Тоді $h(C^i \cdot x_0)$ буде рівнянням-наслідком, але вже меншого степеня. Переконаємось у цьому: якщо $h \in \mathrm{Ann}\,\langle f + 1\rangle$, то $h(C^i \cdot x_0) \cdot (f(C^i \cdot x_0) + 1) = 0$, звідки $h(C^i \cdot x_0) = 0$. Таким чином, якщо виконується один з випадків $g \in \langle A\rangle$ або $h \in \langle f + 1\rangle$, де $\deg g, \deg h < d$, то можна побудувати рівняння-наслідки меншого степеня, тобто провести алгебраїчну атаку на початковий стан генератора гами.

Таким чином, задача пошуку рівнянь-наслідків меншого степеня зводиться до задачі пошуку функцій мінімального степеня в ідеалі $\langle f\rangle$ (або $\langle f + 1\rangle$). Для пошуку функцій мінімального степеня можна скористатись теоремою Арса-Фожере [7], яка стверджує, що функції з найменшим степенем в мінімальному базисі Грьобнера [6] ідеалу $\langle f\rangle$ мають найменший степінь серед усіх функцій в ідеалі $\langle f\rangle$. Отже, задача пошуку функцій мінімального степеня зводиться до задачі побудови мінімального базису Грьобнера ідеалу. Для пошуку базису Грьобнера існує ряд алгоритмів. Найвідомішим прикладом є алгоритм Бухбергера, складність якого в загальному випадку є супер-експоненційною відносно кількості змінних функції f [8]. Є більш сучасні алгоритми пошуку базису Грьобнера, такі як F4 [9] та F5 [10], але в загальному випадку складність цих алгоритмів є експоненційною [11].

Отже, метод пошуку рівнянь-наслідків меншого степеня надає змогу провести атаку на початковий стан потокового шифру. Ця атака є алгебраїчною, оскільки базується на пошуку функцій меншого степеня з

ідеалу $\langle f \rangle$ або $\langle f + 1 \rangle$, тобто використовує алгебраїчні властивості функції f. Зауважимо, що навіть якщо така атака є успішною, то в більшості випадків не вдається побудувати систему лінійних рівнянь і виникає ситуація, коли необхідно розв'язувати систему поліноміальних рівнянь.

Одним зі способів розв'язку системи поліноміальних рівнянь є використання методу *лінеаризації*. Цей метод полягає у виключенні з системи мономів, які містять декілька змінних, шляхом введення нових штучних змінних та рівнянь. Після того, як було побудовано лінеаризовану систему, можна використати метод Гауса для пошуку розв'язків цієї системи. В результаті застосування методу Гауса буде отримано нову систему рівнянь, для якої необхідно виконати повний перебір значень, який в загальному випадку є експоненційним [12].

Варіацією такого методу є алгоритм XL [13], запропонований Ніколасом Куртуа. В цьому алгоритмі фіксується певне число $D > d$ (як правило, в якості D обирається $d + 1$) та формується множина мономів M, степінь яких не перевищує $D - d$ (ця множина включає також константу 1). Потім всі рівняння помножуються на кожний моном з множини M, тому в результаті цього перетворення система рівнянь має розмір $m \cdot |M|$. Отримана система лінеаризується та до неї застосовується метод Гауса. Оскільки кількість рівнянь в системі збільшується, то в деяких випадках такий підхід до розв'язку системи виявляється більш ефективним у порівнянні з традиційним методом лінеаризації.

Ще одним алгоритмом, запропонованим Ніколасом Куртуа, є ElimLin [14]. Цей алгоритм застосовується до системи квадратичних рівнянь. Таким чином, систему поліноміальних рівнянь спочатку необхідно перетворити у систему квадратичних рівнянь за допомогою введення нових змінних та рівнянь. Алгоритм полягає в ітеративному повторенні таких перетворень.

1) Система квадратичних рівнянь впорядковується за довжиною мономів та до неї застосовується метод Гауса, щоб спробувати позбутись від нелінійних мономів. В результаті всі рівняння отриманої системи

можна поділити на лінійні, які формують множину $S_{L'}$, та нелінійні, які формують множину S_T. Якщо множина нелінійних рівнянь є порожньою, то алгоритм зупиняється.

2) В кожному рівнянні з $S_{L'}$ обирається деякий моном, виражається через інші змінні та підставляється в усі рівняння системи. Після такої підстановки лінійне рівняння записується в множину S_L та увесь процес ітеративно повторюється.

Якщо отримана в результаті роботи алгоритму множина лінійних рівнянь S_L включає достатню кількість рівнянь, то можна знайти розв'язок цієї системи, а отже і розв'язок початкової квадратичної системи. В інакшому випадку алгоритм ElimLin не спрацьовує.

В алгоритмах XL та ElimLin основними методами розв'язку є побудова рівнянь-наслідків та збільшення кількості рівнянь в системі. Існують принципово інші алгоритми розв'язку систем поліноміальних рівнянь, наприклад, алгоритм Радума-Сємаєва [15]. В алгоритмі Радума-Сємаєва рівняння в системі записується не у вигляді поліному від багатьох змінних, а у вигляді набору бітових рядків, які є розв'язками цього рівняння та називаються *конфігураціями*. Усі рівняння системи в такому представленні формують граф конфігурацій, в якому кожна вершина відповідає деякому рівнянню, тобто набору конфігурацій для цього рівняння. Далі до цього графу застосовується *алгоритм передачі повідомлень*, в результаті якого в графі залишаються лише ті конфігурації, які не суперечать одна одній, тобто розв'язки системи рівнянь. Можливі ситуації, коли на початку роботи алгоритму конфігурації вже знаходяться в узгодженому стані, тому на цей випадок запропоновано набір додаткових стратегій, які виводять конфігурації з узгодженого стану.

В якості ще одного алгоритму, який використовує принципово інший підхід до розв'язку системи рівнянь, можна навести Zhuang-Zi алгоритм [16]. В цьому алгоритмі задача розв'язку системи поліноміальних рівнянь з багатьма змінними зводиться до задачі

розв'язку системи рівнянь з однією змінною, але над деяким розширенням поля $\mathbb{F}_2$.

Окремим підходом до розв'язку систем поліноміальних рівнянь є використання SAT-розв'язувачів [17]. Як правило, такий підхід застосовується до системи квадратичних рівнянь, тому систему поліноміальних рівнянь необхідно перетворити на систему квадратичних рівнянь шляхом введення нових змінних та рівнянь. Для системи квадратичних рівнянь над $\mathbb{F}_2$, яка містить n змінних та m рівнянь, опишемо принцип побудови еквівалентної формули φ, яка задана у КНФ [18].

1) Оскільки КНФ не містить констант, то для константи 1 необхідно ввести додаткову штучну змінну. Ця змінна має міститись в φ у вигляді окремого диз'юнкту, щоб її єдиним можливим значенням було 1.

2) Для кожного квадратичного монома виду $x_i \cdot x_j$, де $1 \leq i < j \leq n$, виконати заміну $x_{ij} = x_i \cdot x_j$. Оскільки рівність $x_{ij} = x_i \cdot x_j$ еквівалентна рівності $(x_i \vee \overline{x}_{ij}) \wedge (x_j \vee \overline{x}_{ij}) \wedge (\overline{x}_i \vee \overline{x}_j \vee x_{ij}) = 1$, то додати у формулу φ відповідні диз'юнкти. Після цього кроку в системи рівнянь залишаються лише лінійні доданки, тобто система стає лінійною.

3) Для кожного лінійного рівняння $x_1 + x_2 + \ldots + x_l = 0$ побудувати відповідну йому формулу у КНФ. Для цього можна за таблицею істинності булевої формули $x_1 + x_2 + \ldots + x_{l-1}$ побудувати КНФ з 2^{l-1} диз'юнктами, а потім в кожний диз'юнкт додати x_l або $\overline{x}_l$ в залежності від того, яке значення формула $x_1 + x_2 + \ldots + x_l$ приймає на відповідному наборі вхідних значень. Для кожного лінійного рівняння необхідно додати відповідний цьому рівнянню набір диз'юнктів розміру 2^{l-1} в формулу φ.

Розглянемо яку кількість диз'юнктів додає кожний з етапів побудови формули. При заміні одного квадратичного монома необхідно додати три диз'юнкти в формулу, тому загальна кількість таких диз'юнктів буде обмежена поліномом від розміру початкової системи. При перетворенні лінійного рівняння $x_1 + x_2 + \ldots + x_l = 0$ необхідно додати експоненційну від l кількість диз'юнктів. Цього можна запобігти, якщо

розбивати таку суму на набір часткових сум, які пов'язані одна з одною спільними змінними, та розглядати їх як окремі рівняння [12].

Таким чином, систему квадратичних рівнянь над $\mathbb{F}_2$ можна представити у вигляді булевої формули у КНФ. Це надає можливість застосувати до неї відомі SAT-розв'язувачі [19].

Отже, існує багато методів розв'язку системи поліноміальних рівнянь та, відповідно, способів відновлення початкового стану фільтрувального генератора гами, тобто невідомого вектору x_0. В загальному випадку ці всі методи потребують експоненційного обчислення та спрацьовують на практиці лише в часткових випадках, тому виникає необхідність у пошуку альтернативних моделей, в рамках яких можна спробувати відновити невідомий вектор x_0. Такою моделлю може бути відновлення невідомого вектору x_0 за наявністю деякої *часткової інформації* про цей вектор. Ця інформація може бути отримана з побічного каналу зв'язку або з особливостей конкретної реалізації криптосистеми. Оскільки інформація є частковою, то вона не стосується безпосередньо вектору x_0, але може, наприклад, накладати обмеження на значення деяких лінійних залежностей з невідомим вектором x_0. Розглянемо декілька ситуацій, в яких можна отримати таку часткову інформацію.

У випадку фільтрувального генератора гами оновлення стану генератора відбувається за допомогою множення початкового вектора x_0 на супроводжуючу матрицю C. В результаті за m тактів роботи регістру стан генератора приймає значення $C^i \cdot x_0$ для $i = \overline{1, m}$. В цьому випадку частковою інформацією можуть бути деякі значення, які вектори виду $C^i \cdot x_0$, $i = \overline{1, m}$, не можуть приймати.

Аналогічні конструкції зустрічаються в криптосистемах на лінійних кодах. В цих криптосистемах в процесі шифрування виконуються матричні операції з відкритим текстом та ключем. Аналогічно прикладу з початковим станом генератору гами, існує можливість отримати певні обмеження на лінійні залежності, які з'являються під час виконання цих

матричних операцій.

З огляду на наведені модельні припущення та приклади, виникає *задача відновлення невідомого вектору за частковою інформацією, представленою у формі лінійних залежностей.* Для того, щоб мати можливість застосовувати до сформульованої задачі наявні теоретико-множинні конструкції, необхідно формалізувати цю задачу за допомогою окремого математичного об'єкта або навіть теорії. В даній роботі пропонується один з можливих способів такої формалізації введенням понять *лінійної заборони* та *системи лінійних заборон.*

1.2 Необхідні означення та твердження з теорії складності

Розглянемо основні відомості з теорії складності, які будуть застосовуватись під час дослідження систем лінійних заборон.

Задача — це деяке загальне питання, на яке необхідно знайти відповідь. Задача має складатись з двох частин: набір параметрів або вільних змінних та набір властивостей, яким має задовольняти відповідь на цю задачу, щоб вважати таку відповідь розв'язком. *Індивідуальна задача* або *екземпляр задачі* — це задача, в якій всі параметри зафіксовані конкретними значеннями [20].

Задачею розпізнавання будемо називати задачу, в якій дві можливих відповіді — «Так» або «Ні». Таким чином, набір усіх вхідних даних або індивідуальних задач можна однозначно класифікувати на дві групи в залежності від відповіді. *Задачею пошуку* будемо називати задачу, в якій необхідно знайти об'єкт, що задовольняє умовам цієї задачі.

Алгоритмом будемо називати чітко визначену послідовність дій. Під імовірнісним алгоритмом буде розуміти алгоритм, який має доступ до джерела випадковості, тобто генератора випадкових об'єктів (надалі це будуть переважно елементи скінченного поля). Генератор, як правило, буде мати рівноймовірний розподіл на множині елементів, тобто всі елементи цієї множини будуть обиратись з однаковою імовірністю.

Складністю операції генерування випадкових об'єктів будемо нехтувати.

Для того, щоб визначити складність задачі, можна довести приналежність цієї задачі до певного класу складності. Надамо означення основним класам складності за часом $\mathcal{P}$ і $\mathcal{NP}$ [21].

Будемо казати, що алгоритм B розв'язує певну задачу розпізнавання Π, якщо $B(x) = 1 \Leftrightarrow \Pi(x) = 1$. Алгоритм B розв'язує задачу за поліноміальний час, якщо на вході x час роботи алгоритму (або кількість елементарних операцій) не перевищує $\text{poly}(|x|)$, де $\text{poly}(|x|)$ — це поліном, степінь якого є константою.

Означення 1.1. Задача розпізнавання *належить класу складності* $\mathcal{P}$, якщо існує алгоритм, який розв'язує цю задачу за поліноміальний від довжини входу час.

Таким чином, клас складності $\mathcal{P}$ складається з усіх задач розпізнавання, для яких існує ефективний розв'язок.

Визначимо алгоритм верифікації V як алгоритм, який приймає на вхід два аргументи, перший з яких — це вхідні дані задачі x, а другий — це бінарний рядок y, який називається сертифікатом. Алгоритм V верифікує вхідний рядок x, якщо існує сертифікат y такий, що $V(x, y) = 1$. Аналогічно, алгоритм V є поліноміальним, якщо час роботи (або кількість елементарних операцій) є поліномом від довжини входу.

Будемо казати, що алгоритм V верифікує задачу розпізнавання Π за поліноміальний час, якщо цей алгоритм є поліноміальним та довжина сертифікату обмежена довжиною полінома. Інакше кажучи, алгоритм V верифікує Π за поліноміальний час, якщо для кожного входу $x \in \{0, 1\}^*$ існує сертифікат y, довжина якого обмежена поліномом від довжини вхідних даних, такий що $\Pi(x) = 1 \Leftrightarrow V(x, y) = 1$, причому час роботи V є поліноміальним від довжини входу.

Означення 1.2. Задача розпізнавання *належить класу складності* $\mathcal{NP}$, якщо існує алгоритм, який верифікує цю задачу за поліноміальний від довжини входу час.

Таким чином, клас складності $\mathcal{NP}$ складається з усіх задач розпізнавання, розв'язок яких можна ефективно перевірити.

Очевидно, що $\mathcal{P} \subseteq \mathcal{NP}$, тому що знайдений розв'язок є автоматично перевіреним. Питання чи $\mathcal{NP} \subseteq \mathcal{P}$ залишається відкритим.

Розглянемо різні типи звідності між задачами. Спочатку розглянемо поліноміальну звідність або звідність за Карпом [22].

Означення 1.3. Задача розпізнавання Π_1 *поліноміально зводиться* до задачі розпізнавання Π_2, позначається $\Pi_1 \leq_p \Pi_2$, якщо існує функція $f : \{0,1\}^* \to \{0,1\}^*$, яка обчислюється за поліноміальний час, та $\forall x \in \{0,1\}^*$ виконується: $\Pi_1(x) = 1 \Leftrightarrow \Pi_2(f(x)) = 1$.

Якщо для задач Π_1 та Π_2 виконується $\Pi_1 \leq_p \Pi_2$ та $\Pi_2 \leq_p \Pi_1$, то будемо називати такі задачі *еквівалентними відносно поліноміальної звідності* та позначати $\Pi_1 =_p \Pi_2$.

Розглянемо звідність за Тюрінгом (інша назва — звідність за Куком) [20].

Означення 1.4. Задача розпізнавання Π_1 *зводиться за Тюрінгом* до задачі розпізнавання Π_2, позначається $\Pi_1 \leq_T \Pi_2$, якщо існує поліноміальний алгоритм A, який за поліноміальну кількість звернень до оракула, що розв'язує задачу Π_2, розв'язує задачу Π_1.

Зауваження. Це означення має місце не тільки для задач розпізнавання, а й для задач пошуку, причому зводитись одна до одної можуть задачі різних типів.

Зведення за Карпом можна розглядати як частковий випадок зведення за Тюрінгом, в якому можна зробити лише один запит до оракула Π_2 та необхідно повернути саме ту відповідь, яку цей оракул надав.

Якщо для задач Π_1 та Π_2 виконується $\Pi_1 \leq_T \Pi_2$ та $\Pi_2 \leq_T \Pi_1$, то будемо називати такі задачі *еквівалентними відносно звідності по Тюрінгу* та позначати $\Pi_1 =_T \Pi_2$.

Задача П називається $\mathcal{NP}$-*складною*, якщо для будь-якої задачі $\Pi' \in \mathcal{NP}$ виконується $\Pi' \leq_p \Pi$. Таким чином, задача є $\mathcal{NP}$-складною, якщо будь-яка інша задача з класу $\mathcal{NP}$ поліноміально зводиться до цієї задачі.

Якщо задача П належить класу $\mathcal{NP}$ і є $\mathcal{NP}$-складною, то будемо називати її $\mathcal{NP}$-*повною*. $\mathcal{NP}$-повні задачі формують множину найскладніших задач у класі $\mathcal{NP}$.

Для того, щоб довести, що деяка задача є $\mathcal{NP}$-повною, необхідно показати, що:

1) ця задача належить класу $\mathcal{NP}$;

2) деяка інша $\mathcal{NP}$-повна задача поліноміально зводиться до неї.

Таким чином, маючи хоча б одну $\mathcal{NP}$-повну задачу, можна доводити $\mathcal{NP}$-повноту інших задач користуючись наведеним фактом та транзитивністю поліноміального зведення.

Існування $\mathcal{NP}$-повної задачі — це суто технічний факт, приклад такої $\mathcal{NP}$-повної задачі можна знайти у [22]. Більш нетривіальним є пошук $\mathcal{NP}$-повної задачі, яка була б застосовна на практиці. Така задача була знайдена Куком у 1971 році та отримала назву SAT [23]. Ця задача зіграла фундаментальну роль в розвитку теорії складності, оскільки за допомогою неї вдалось довести $\mathcal{NP}$-повноту багатьох задач, які використовуються на практиці. Перший вагомий список $\mathcal{NP}$-повних задач з'явився у роботі Карпа, яка була опублікована в 1972 році [24].

Розглянемо задачу SAT (формулювання взято з [22, 25]).

Задача 1.1 (SAT)**.** *Вхід.* Булева формула у КНФ ϕ з n змінними та m диз'юнктами.

Необхідно з'ясувати чи існує такий набір $(x_1, x_2, \ldots, x_n)$, що $\phi(x_1, x_2, \ldots, x_n) = 1$.

Вихід. «Так», якщо такий набір існує, «Ні» інакше.

Часто формулюють SAT у випадку, коли кількість літералів (літералом будемо називати змінну або її заперечення) в одному диз'юнкті не перевищує деякого зафіксованого числа.

Задача 1.2 (kSAT)**.** *Вхід.* Булева формула у КНФ ϕ з n змінними та m диз'юнктами, де кожний з диз'юнктів містить не більше k літералів.

Необхідно з'ясувати чи існує такий набір $(x_1, x_2, \ldots, x_n)$, що $\phi(x_1, x_2, \ldots, x_n) = 1$.

Вихід. «Так», якщо такий набір існує, «Ні» інакше.

Відомо, що 2SAT належить класу $\mathcal{P}$, тобто для цієї задачі існує ефективний алгоритм розв'язку, а всі kSAT для $k \geq 3$ є $\mathcal{NP}$-повними задачами.

Іншою варіацією SAT є задача, в якій на вхід надходить булева формула, але всі операції диз'юнкції в диз'юнктах замінені на XOR. Для зручності, такі модифіковані диз'юнкти будемо називати XOR-виразами.

Задача 1.3 (XORSAT)**.** *Вхід.* Булева формула ϕ з n змінними та m XOR-виразами.

Необхідно з'ясувати чи існує такий набір $(x_1, x_2, \ldots, x_n)$, що $\phi(x_1, x_2, \ldots, x_n) = 1$.

Вихід. «Так», якщо такий набір існує, «Ні» інакше.

Відомо, що XORSAT належить класу $\mathcal{P}$ — для ефективного розв'язку цієї задачі можна скористатись методом Гауса розв'язку систем лінійних рівнянь.

В контексті аналізу складності систем лініних заборон важливу роль буде відігравати модифікація задачі XORSAT, в якій необхідно встановити, що лише деякий набір XOR-виразів приймає значення 1, не обов'язково всі вирази одночасно.

Задача 1.4 (Max-XORSAT)**.** *Вхід.* Булева формула ϕ з n змінними та m XOR-виразами, число l, де $1 \leq l \leq m$.

Необхідно з'ясувати чи існує такий набір $(x_1, x_2, \ldots, x_n)$, що прийнамні l XOR-виразів набувають значення 1.

Вихід. «Так», якщо такий набір існує, «Ні» інакше.

Відомо, що задача Max-XORSAT є $\mathcal{NP}$-повною. Якщо розглянути обмежену версію цієї задачі — Max-kXORSAT, то вона буде $\mathcal{NP}$-повною

починаючи з $k \geq 2$ [25]. Для доведення $\mathcal{NP}$-повноти Max-XORSAT можна обрати задачу Max-Cut, яка використовується для доведення $\mathcal{NP}$-повноти широкого класу задач.

Ще одним важливим прикладом $\mathcal{NP}$-повної задачі є задача існування розв'язку системи квадратичних рівнянь над полем $\mathbb{F}_{2^k}$.

Задача 1.5 (MQ). *Вхід.* $\mathbb{F}_{2^k}$, поліноми $p_1, p_2, \ldots, p_m$ над полем $\mathbb{F}_{2^k}$, $k \geq 1$, кожний з поліномів приймає на вхід n змінних та $\deg p_i \leq 2$ для $i = \overline{1, m}$.

Необхідно з'ясувати чи існує набір $(x_1, x_2, \ldots, x_n)$ такий, що $p_i(x_1, x_2, \ldots, x_n) = 0$ для $i = \overline{1, m}$.

Вихід. «Так», якщо такий набір існує, «Ні» інакше.

$\mathcal{NP}$-повнота задачі MQ для випадку $\mathbb{F} = \mathbb{F}_{2^k}$ була доведена в роботі [26], а у випадку довільного поля $\mathbb{F}$ в роботі [27]. Для доведення $\mathcal{NP}$-повноти в останній роботі використовувалась техніка арифметизації: при зведенні задачі 3SAT до MQ всі диз'юнкти в булевій формулі записувались у вигляді поліномів Жегалкіна. Степінь таких поліномів у випадку 3SAT не може перевищувати трьох. З отриманих виразів формувалась система кубічних рівнянь з багатьма змінними відносно XOR. Для позбавлення від кубічних мономів застосовувалась схема, запропонована Л. Валіантом, яка полягала у виключенні кубічних мономів шляхом введення нових змінних та рівнянь в систему.

Задача MQ має широкий спектр застосування. В криптографії MQ використовується для побудови криптосистем: криптосистема SimpleMatrix [28], системи цифрового підпису UOV [29] та Rainbow [30]. MQ також застосовується в алгебраїчному криптоаналізі потокових шифрів, детальніше таке застосування MQ досліджується в підрозділі 1.1. Проводяться дослідження у напряму ефективного розв'язку часткових випадків цієї задачі в залежності від виду вхідної системи [31, 32].

Іншим важливим класом складності задач є імовірнісний клас $\mathcal{RP}$ [22]. Для того, щоб визначити цей клас, необхідно сформулювати означення імовірнісного алгоритму з односторонньою помилкою. Будемо

казати, що задача розпізнавання Π розв'язується імовірнісним алгоритмом R з односторонньою помилкою, коли виконуються такі умови:

1) якщо $\Pi(x) = 1$, то $R(x) \geq \frac{1}{2}$;

2) якщо $\Pi(x) = 0$, то $R(x) = 0$.

Імовірнісний алгоритм R називається поліноміальним, якщо час його роботи обмежений поліномом від довжини вхідних даних.

Означення 1.5. Задача розпізнавання Π належить *класу складності* $\mathcal{RP}$, якщо для неї існує поліноміальний імовірнісний алгоритм з односторонньою помилкою.

Відомо, що клас $\mathcal{RP} \subseteq \mathcal{NP}$. Аналогічно можна визначити клас складності задач $co\mathcal{RP}$, для яких існує імовірнісний алгоритм с односторонньою помилкою, але ця помилка відбувається у випадку $\Pi(x) = 0$. Прикладом задачі з класу складності $co\mathcal{RP}$ є задача, в якій необхідно з'ясувати чи дорівнює поліном тотожно нулю, де поліном задано над $\mathbb{Z}$. Побудова імовірнісного алгоритму з односторонньою помилкою для цієї задачі використовує лему Шварца-Зіпеля [33], яка буде також використовуватись в контексті систем лінійних заборон.

Висновки до розділу 1

В цьому розділі було розглянуто основні методи алгебраїчного криптоаналізу. Більшість цих методів спрямована на пошук розв'язку системи поліноміальних рівнянь, оскільки система поліноміальних рівнянь є одним з основних предметів дослідження алгебраїчного криптоаналізу. Запропоновано розглянути більш широкий клас задач, які спрямовані на відновлення невідомого вектору за частковою інформацією. Висунуто модельні припущення про вигляд цієї інформації та сформульовано задачу відновлення вектора за частковою інформацією, представленої у формі лінійних залежностей.

Також розглянуто основні факти з теорії складності, необхідні для подальших досліджень. Вони охоплюють означення класів складності задач та типів звідності між задачами. Розглянуто $\mathcal{NP}$-повні задачі, які відіграють важливу роль в дослідженні систем лінійних заборон, та оцінки складності задач, пов'язаних з системами лінійних заборон.

2 АЛГЕБРАЇЧНІ ТА КОМБІНАТОРНІ ВЛАСТИВОСТІ СИСТЕМ ЛІНІЙНИХ ЗАБОРОН НАД СКІНЧЕННИМ ПОЛЕМ

В даному розділі формалізується задача відновлення невідомого вектора за частковою інформацією, представленою у формі лінійних залежностей, шляхом введення нотації системи лінійних заборон над скінченним полем. Система лінійних заборон розглядається як окремий математичний об'єкт, тому виникає необхідність побудови теорії, яка б містила в собі змістовні твердження стосовно систем лінійних заборон. Побудова такої теорії потребує встановлення зв'язків з наявною математичною базою знань, тому більша частина цього розділу присвячена пошуку таких зв'язків. В результаті цих досліджень доводиться ряд алгебраїчних та комбінаторних властивостей системи лінійних заборон.

2.1 Означення та властивості систем лінійних заборон

Визначимо лінійну заборону та систему лінійних заборон за аналогією до лінійного рівняння та системи лінійних рівнянь.

Означення 2.1. *Лінійною забороною* над полем $\mathbb{F}_{2^k}$ будемо називати вираз типу

$$a_1x_1 + a_2x_2 + \cdots + a_xx_n \neq a_0,$$

де $a_i \in \mathbb{F}_{2^k}$ для $i = \overline{0,n}$, $x_i \in \mathbb{F}_{2^k}$ для $i = \overline{1,n}$.

Якщо позначити $a = (a_1, a_2, \ldots, a_n)$, то можемо переписати лінійну заборону у вигляді виразу $(a,x) \neq a_0$, де $(a,x) := \sum_{i=1}^{n} a_ix_i$ — це скалярний добуток векторів a та x. Фактично, лінійна заборона означає, що $a_1x_1 + a_2x_2 + \ldots + a_nx_n \in \mathbb{F}_{2^k} \setminus \{a_0\}$, а це еквівалентно такій

сукупності рівнянь:

$$\begin{cases} a_1x_1 + a_2x_2 + \ldots + a_nx_n = a'_1 \\ \ldots \\ a_1x_1 + a_2x_2 + \ldots + a_nx_n = a'_l \end{cases},$$

де $a'_i \in \mathbb{F}_{2^k} \setminus \{a_0\}$ для $i = \overline{1,n}$ та $l = 2^k - 1$.

Зауваження. В залежності від контексту будемо також називати лінійною забороною вектор $(a_1, a_2, \ldots, a_n, a_0)$ довжини $n+1$ у випадку $a_0 \neq 0$ та вектор $(a_1, a_2, \ldots, a_n)$ довжини n у випадку $a_0 = 0$.

Зауваження. Якщо для $x_0 \in \mathbb{F}_{2^k}^n$ виконується $(a, x_0) = 0$, то будемо казати, що вектор a *забороняє* x_0. Також в такому разі будемо казати, що вектори a та x_0 є *ортогональними*.

Означення 2.2. *Розв'язком лінійної заборони* будемо називати такий вектор $x_0 \in \mathbb{F}_{2^k}^n$, що $(a, x_0) \neq a_0$. *Множиною розв'язків лінійної заборони* будемо називати множину векторів $D = \{x \in \mathbb{F}_{2^k}^n | (a, x) \neq a_0\}$. Якщо множини розв'язків двох лінійних заборон збігаються, то будемо називати такі лінійні заборони *еквівалентними*.

Маючи означення множини розв'язків лінійної заборони виникає питання які елементарні дії над заборонами можна виконувати, отримуючи при цьому еквівалентні заборони. Необхідно перевірити чи виконуються для лінійних заборон елементарні перетворення лінійних рівнянь, такі як перенесення доданків з однієї частини в іншу та множення обох частин на константу.

Твердження 2.1. *Множина розв'язків лінійної заборони не змінюється при перенесенні доданків з однієї частини в іншу, а також при множенні обох частин на константу, яка не дорівнює нулю.*

Доведення. Розглянемо лінійну заборону

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \neq a_0.$$

Позначимо $D_0 = \{x \in \mathbb{F}_{2^k} | (a, x) \neq a_0\}$, $D_0' = \mathbb{F}_{2^k} \setminus D_0$, тобто D_0' складається з $x \in \mathbb{F}_{2^k}^n$ таких, що $(a, x) = 0$.

Сформуємо іншу лінійну заборону

$$a_2x_2 + \ldots + a_nx_n \neq a_1x_1 + a_0.$$

Аналогічно визначимо множини D_1 та D_1'. Множина D_1' складається з усіх $x \in \mathbb{F}_{2^k}^n$ таких, що $a_2x_2 + \ldots + a_nx_n = a_1x_1 + a_0$. Перенесемо доданок a_1x_1 у ліву частину рівняння: $a_1x_1 + a_2x_2 + \ldots + a_nx_n = a_0$. Звідси випливає, що $D_1' = D_0'$, а отже $D_1 = D_0$, оскільки множини D_0, D_0' та D_1, D_1' не перетинаються.

При множенні на константу доведення є аналогічним. □

Оскільки поле $\mathbb{F}_{2^k}$ є скінченною множиною, то можемо обчислити потужність множини розв'язків лінійної заборони.

Твердження 2.2. *Нехай $a_1x_1 + a_2x_2 + \ldots + a_nx_n \neq a_0$ — лінійна заборона над полем $\mathbb{F}_{2^k}$, тоді $|D| = 2^{kn} - 2^{k(n-1)}$.*

Доведення. Розглянемо відповідне до лінійної заборони лінійне рівняння $a_1x_1 + a_2x_2 + \ldots + a_nx_n = a_0$. Якщо маємо справу з нетривіальним випадком $a \neq \overline{0}$, то деякий з $a_1, a_2, \ldots, a_n$ є ненульовим елементом поля; не обмежуючи загальності припустимо, що це x_n. Тоді можемо переписати рівняння в наступному еквівалентному вигляді:

$$a_n^{-1} \cdot (a_1x_1 + a_2x_2 + \ldots + a_{n-1}x_{n-1} + a_0) = x_n$$

Пробігаючи усі можливі вектори $(x_1, x_2, \ldots, x_{n-1})$ отримуємо деякі значення x_n такі, що $(x_1, x_2, \ldots, x_n)$ є розв'язком лінійного рівняння.

Якщо D' — це множина розв'язків цього лінійного рівняння, то

$$|D'| = |\{(x_1, x_2, \ldots, x_{n-1}), x_i \in \mathbb{F}_{2^k}, i = \overline{1, n}\}| = 2^{k(n-1)}.$$

Тоді $D = \mathbb{F}_{2^k} \setminus D'$, а отже

$$|D| = |\mathbb{F}_{2^k}^n| - |D'| = 2^{kn} - 2^{k(n-1)}.$$

Слід зауважити, що інший спосіб отримати цей результат — помітити, що множина розв'язків відповідного лінійного рівняння формує лінійний підпростір розмірності $n-1$, потужність якого дорівнює $2^{k(n-1)}$. □

Окремо розглянемо випадок лінійної заборони при $n = 1$, тобто $a \cdot x \neq a_0$, де $a \neq 0$. Якщо $a_0 = 0$, то розв'язками будуть усі $x \in \mathbb{F}_{2^k}^*$, кількість яких складає $2^k - 1$. Якщо $a_0 \in \mathbb{F}_{2^k}^*$, то множиною розв'язків буде $\{z \in \mathbb{F}_{2^k}, z \neq a_0 \cdot a^{-1}\}$; її потужність теж становить $2^k - 1$. Отже, у випадку $n = 1$ кількість розв'язків лінійної заборони узгоджується з формулою $2^{kn} - 2^{k(n-1)}$.

Означення 2.3. Системою лінійних заборон будемо називати систему співвідношень виду:

$$\begin{cases} a_1^{(1)}x_1 + a_2^{(1)}x_2 + \ldots + a_n^{(1)}x_n \neq a_0^{(1)} \\ a_1^{(2)}x_1 + a_2^{(2)}x_2 + \ldots + a_n^{(2)}x_n \neq a_0^{(2)} \\ \qquad\qquad \ldots \\ a_1^{(m)}x_1 + a_2^{(m)}x_2 + \ldots + a_n^{(m)}x_n \neq a_0^{(m)} \end{cases},$$

де $a_i^{(j)} \in \mathbb{F}_{2^k}$ для $i = \overline{0,n}, j = \overline{1,m}$, $x_i \in \mathbb{F}_{2^k}$ для $i = \overline{1,n}$ та $m > 1$.

Більш компактний запис: будемо позначати $a^{(j)} := (a_1^{(j)}, a_2^{(j)}, \ldots, a_n^{(j)})$, тоді систему лінійних заборон можна переписати у вигляді:

$$\begin{cases} (a^{(1)}, x) \neq a_0^{(1)} \\ (a^{(2)}, x) \neq a_0^{(2)} \\ \qquad \ldots \\ (a^{(m)}, x) \neq a_0^{(m)} \end{cases},$$

де $a^{(j)} \in \mathbb{F}_{2^k}^n$ для $j = \overline{1,m}$ та $x \in \mathbb{F}_{2^k}^n$.

Ще більш компактний запис: нехай A — матриця розміру $m \times n$ над полем $\mathbb{F}_{2^k}$ виду

$$A = \begin{pmatrix} a^{(1)} \\ a^{(2)} \\ \dots \\ a^{(m)} \end{pmatrix} = \begin{pmatrix} a_1^{(1)} & a_2^{(1)} & \dots & a_n^{(1)} \\ a_1^{(2)} & a_2^{(2)} & \dots & a_n^{(2)} \\ & & \dots & \\ a_1^{(m)} & a_2^{(m)} & \dots & a_n^{(m)} \end{pmatrix} = \{a_i^{(j)}\}_{i=\overline{1,n}}^{j=\overline{1,m}}.$$

Тоді систему лінійних заборон можна записати у вигляді:

$$A \cdot x \neq a_0, \text{ де } a_0 = \begin{pmatrix} a_0^{(1)} \\ a_0^{(2)} \\ \dots \\ a_0^{(m)} \end{pmatrix}.$$

Зауваження. Запис $A \cdot x \neq a_0$ є умовним та вводиться для компактності подальших формул. Загальноприйнятий запис $a \neq b$ для деяких векторів a та b означає, що ці вектори відрізняються в деякій координаті, в нашому ж випадку він означає, що вони відрізняються в усіх координатах.

Аналогічно до лінійної заборони можемо визначити розв'язок системи лінійних заборон.

Означення 2.4. *Розв'язком системи лінійних заборон* будемо називати $x_0 \in \mathbb{F}_{2^k}^n$ такий, що $A \cdot x_0 \neq a_0$. *Множиною розв'язків системи лінійних заборон* будемо називати множину векторів

$$\widetilde{D} = \{x \in \mathbb{F}_{2^k}^n | A \cdot x \neq a_0\} = D_1 \cap D_2 \cap \ldots \cap D_m,$$

де D_j — це множина розв'язків відповідної лінійної заборони в системі,

$|D_j| = 2^{kn} - 2^{k(n-1)}$, $j = \overline{1,m}$. Дві системи лінійних заборон будемо називати *еквівалентними*, якщо в них збігаються множини розв'язків.

Твердження 2.3. *Множина розв'язків системи лінійних заборон не змінюється при перенесенні доданків з однієї частини в іншу в будь-якій з заборон та при множенні обох частин будь-якої заборони на константу, яка не дорівнює нулю.*

Доведення. Нехай в деякій забороні виконали або перенесення доданку або множення на константу. Тоді за твердженням 2.1 множина розв'язків цієї заборони не змінилась, а отже не змінився перетин усіх множин розв'язків лінійних заборон. □

Зауваження. Для спрощення обчислень в деяких випадках будемо розглядати систему лінійних заборон з нульовими правими частинами. Також покладемо $\mathbb{F} = \mathbb{F}_{2^k}$, хоча у випадку $\mathbb{F}$ — довільного скінченного поля та нульових правих частин більшість результатів буде виконуватись, якщо ні, то про це буде окремо сказано.

2.2 Критерій існування розв'язків систем лінійних заборон

Розглянемо критерій існування розв'язку систем лінійних заборон над скінченним полем. Цей критерій пов'язує арифметику систем лінійних заборон з арифметикою поліномів та дозволяє використовувати методи, розроблені для поліномів, для вивчення систем лінійних заборон над скінченним полем.

Теорема 2.1. *У системи лінійних заборон $A \cdot x \neq \overline{0}$ над полем $\mathbb{F}_{2^k}$ існує розв'язок тоді і тільки тоді коли*

$$\prod_{i=1}^{m} (a^{(i)}, x) \not\equiv 0.$$

Доведення. Позначимо для зручності $F(x) = \prod_{i=1}^{m} (a^{(i)}, x)$.

Будемо доводити цей факт у наступній формі: у системи лінійних

заборон $A \cdot x \neq \overline{0}$ не існує розв'язків тоді і тільки тоді коли $F(x) \equiv 0$. Оскільки доведення необхідності та достатності цього критерію є досить подібними, то не будемо окремо розглядати необхідність на достатність.

Відсутність розв'язків системи лінійних заборон виду

$$\begin{cases} (a^{(1)}, x) \neq 0 \\ (a^{(2)}, x) \neq 0 \\ \quad \dots \\ (a^{(m)}, x) \neq 0 \end{cases}$$

рівносильно тому, що для кожного $x \in \mathbb{F}_{2^k}^n$ хоча б одна з заборон не виконується, тобто перетворюється на рівність, що у вигляді формули можна записати таким чином:

$$\forall x \in \mathbb{F}_{2^k}^n : \exists i \in \overline{1, m} : (a^{(i)}, x) = 0.$$

Розглянемо поліном, що складається з добутку лівих частин усіх лінійних заборон: $F(x) = (a^{(1)}, x) \cdot (a^{(2)}, x) \cdot \ldots \cdot (a^{(m)}, x)$. Зафіксуємо деякий x, тоді для нього буде існувати співмножник, який дорівнює 0, отже увесь добуток буде перетворюватись на нуль: $F(x) = 0$. Таким чином, $\forall x : F(x) = 0$. А це, фактично, і означає, що поліном F над полем $\mathbb{F}_{2^k}$ тотожно дорівнює нулю, оскільки він обертається в нуль на кожному можливому вході. □

Зауваження. В умові теореми 2.1 в системі лінійних заборон усі праві частини дорівнювали нулю. Якщо вони будуть ненульовими, то це не вплине на хід доведення. В такому разі поліном $F(x)$ буде визначатись таким чином:

$$F(x) = \prod_{i=1}^{m} [(a^{(i)}, x) + a_0^{(i)}].$$

Більш того, цей критерій може бути застосований до довільних функцій у лівих частинах заборон, не обов'язково лінійних. Але в цьому випадку степінь результуючого полінома $F(x)$ вже не буде дорівнювати m, а буде

становити, наприклад $2 \cdot m$, у випадку квадратичних лівих частин.

Зауваження. Цей критерій показує, що в системі лінійних заборон над скінченним полем можна отримувати заборони-наслідки шляхом множення лівих частин. Нехай є два поліноми $f(x)$ та $g(x)$ над скінченним полем, тоді $f(x) \neq 0$ та $g(x) \neq 0$ в тому і тільки в тому разі, коли $f(x) \cdot g(x) \neq 0$. Нескладно в цьому переконатись: якщо $f(x) \neq 0$ та $g(x) \neq 0$, то ці поліноми приймають деякі значення з мультиплікативної групи поля, а добуток двох елементів мультиплікативної групи поля не може вийте за її межі, тому $f(x) \cdot g(x) \neq 0$. В інакшу сторону доведення аналогічне. Тому сформульований критерій можна розглядати як варіант використання цієї властивості, при якому знайшли добуток лівих частин усіх заборон. Оскільки в системі лінійних заборон необхідно знайти такий x, що $f(x) \cdot g(x) \neq 0$, то питання існування такого x приводить до питання тотожної рівності полінома $f(x) \cdot g(x)$ нулю.

Зауваження. Звернемо увагу на те, що наведений критерій не дає можливості конструктивно шукати розв'язки системи лінійних заборон. Це пов'язано з тим, що означення полінома, який не дорівнює тотожно нулю, не потребує явно задавати значення аргументу, на якому цей поліном не обертається в нуль.

З огляду на цей критерій виникає ідея аналітично перевіряти чи дорівнює тотожно нулю поліном $F(x)$. На жаль, при розкритті дужок потенційно може виникнути експоненційна кількість доданків — маємо справу з поліномом степеня m від n змінних, який містить в загальному випадку C_{n+m-1}^{n-1} коефіцієнтів. Цей біноміальний коефіцієнт не можна обмежити деяким поліномом від n та m, тому в такому виді цей критерій не можна використовувати для побудови поліноміального алгоритму для перевірки існування розв'язку системи лінійних заборон або поліноміального зведення питання існування розв'язку системи до задачі визначення чи тотожно дорівнює поліном над деяким скінченним полем нулю. Більш того, навіть розкриття дужок не завжди надає відповідь на це питання, оскільки в скінченному полі існують поліноми з ненульовими

коефіцієнтами, які дорівнюють тотожно нулю.

Але існують імовірнісні тести, які за заданим поліномом визначають чи дорівнює тотожно цей поліном нулю. В цих тестах не потрібно аналітично розкривати усі дужки; достатньо використовувати поліном як «чорну скриньку» та обчислювати значення цього поліному в деяких точках. А обчислення значення поліному вже не є експоненційним обчисленням, навіть коли, як в нашому випадку, обчислюється значення декількох функцій, тобто лівих частин усіх лінійних заборон в системі.

Найвідоміший та водночас найпростіший такий тест засновано на лемі Шварца-Зіпеля, яка у випадку поля $\mathbb{F} = \mathbb{F}_{2^k}$ (але ця лема розповсюджується на скінченну підмножину довільного поля) стверджує, що імовірність по всім $r_1, r_2, \ldots, r_n$, які є елементами поля $\mathbb{F}_{2^k}$, того, що поліном $f(r_1, r_2, \ldots, r_n)$, $\deg f = m$, дорівнює нулю, обмежена зверху величиною $\frac{m}{|\mathbb{F}_{2^k}|}$, інакше кажучи:

$$Pr_{r_1,r_2,\ldots,r_n \in \mathbb{F}_{2^k}}[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{m}{2^k}.$$

Ця лема працює лише для випадку $m \leq 2^k$, оскільки інакше будемо отримувати тривіальну оцінку імовірності, яка не є інформативною. Більше того, якщо обрати $m \leq 2^{k-1}$, то отримаємо класичний випадок, в якому при ітеративному повторенні перевірки умови $f(r_1, r_2, \ldots, r_n) = 0$ помилка буде експоненційно зменшуватись, оскільки вона обмежена константою $\frac{2^{k-1}}{2^k} = \frac{1}{2}$.

Отже, можемо сформулювати імовірнісний алгоритм перевірки того, чи існує розв'язок у системи лінійних заборон у випадку коли $m \leq 2^{k-1}$.

Алгоритм 2.1. *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$, вектор a_0 розміру $m \times 1$.

1) Побудувати поліном $F(x) = \prod_{i=1}^{m}[(a^{(i)}, x) + a_0^{(i)}]$, де $\deg F = m$.

2) Повторювати d разів:

а) Вибрати випадково та рівноймовірно $(r_1, r_2, \ldots, r_n) \in \mathbb{F}_{2^k}^n$.

б) Перевірити чи виконується $F(r_1, r_2, \ldots, r_n) \neq 0$. Якщо виконується, повернути «Так».

3) Повернути «Ні».

Вихід. «Так», якщо система має розв'язок, «Ні», інакше.

Отримуємо тест з односторонньою помилкою.

1) Припустимо на вході система $A \cdot x \neq a_0$, у якої існує розв'язок. Тоді $F(x) \equiv 0$. В такому разі імовірність отримати відповідь «Так» дорівнює нулю, оскільки «Так» повертаємо лише тоді, коли знайшли вектор, на якому поліном не дорівнює нулю, а для тотожно рівного нулю полінома такого значення вектора не існує. В такому разі, імовірність отримати «Ні» дорівнює 1.

2) Припустимо на вході система $A \cdot x \neq a_0$, у якої немає розв'язків. Тоді виконується $F(x) \not\equiv 0$. Всі генерації випадкових наборів $(r_1, r_2, \ldots, r_n)$ є незалежними подіями, тому імовірність того, що поліном приймає значення нуль одночасно на всіх цих наборах, дорівнює добутку імовірностей окремих подій. Тому через d ітерацій маємо оцінку:

$$Pr_{r_1,r_2,\ldots,r_n \in \mathbb{F}_{2^k}}[f(r_1, r_2, \ldots, r_n) = 0] \leq \left(\frac{m}{2^k}\right)^d \leq \left(\frac{1}{2}\right)^d.$$

Отже, можемо оцінити імовірність того, що поліном не дорівнює тотожно нулю:

$$Pr_{r_1,r_2,\ldots,r_n \in \mathbb{F}_{2^k}}[f(r_1, r_2, \ldots, r_n) \neq 0] \geq 1 - \left(\frac{1}{2}\right)^d.$$

У цьому випадку даємо правильну відповідь, тобто «Так», з імовірністю $1 - \left(\frac{1}{2}\right)^d$, а відповідь «Ні» з імовірністю $\left(\frac{1}{2}\right)^d$.

Отримані обчислення можна представити у вигляді матриці помилок 2.1, яка відображає сумісні імовірності усіх гіпотез та дійсних фактів.

Таблиця 2.1 – Матриця помилок алгоритму для d ітерацій

	«Так»	«Ні»
$A \cdot x \neq a_0$ має розв'язок	$\geq 1 - \left(\frac{1}{2}\right)^d$	$\leq \left(\frac{1}{2}\right)^d$
$A \cdot x \neq a_0$ не має розв'язків	0	1

Проаналізуємо складність цього алгоритму. Операція обчислення значення полінома в деякій точці потребує: $m \times n$ операцій множення у полі для обчислення усіх проміжних значень лівих частин системи, а також m операцій множення для знаходження загального добутку. Отримуємо $m \cdot (n+1)$ операцій множення у полі. Цю процедуру необхідно повторити d разів, отже взагалом необхідно $d \cdot m \cdot (n+1) = O(d \cdot m \cdot n)$ операцій, тому наведений алгоритм є поліноміальним імовірнісним. Надалі спрощена версія цього алгоритму буде використана, щоб довести приналежність задачі існування розв'язку системи лінійних заборон (з обмеженням на m) до імовірнісного класу складності $\mathcal{RP}$. Обмеження на m можуть бути зняті [35], це планується у подальших дослідженнях.

Зазначимо, що у часткових випадках з'ясувати чи існують розв'язки системи лінійних заборон можна виключно за видом цієї системи. Наприклад, якщо в системі міститься такий x_i, де $1 \leq i \leq n$, що всі коефіцієнти $a_i^{(j)} \neq 0$ для $j = \overline{1, m}$, то розв'язки в системи гарантовано існують. Вектори, у яких на i-ій позиції міститься ненульовий елемент, а всі інші компоненти дорівнюють нулю, будуть такими розв'язками. В загальному випадку такий x_i для $1 \leq i \leq n$ не завжди існує.

2.3 Властивості розв'язків систем лінійних заборон з нульовими правими частинами

Припустимо, що відомий один з розв'язків системи лінійних заборон з нульовими правими частинами. Виникає питання чи можна в такому разі щось сказати про інші розв'язки чи можливо навіть відновити деякі з них.

Твердження 2.4. *Нехай* $z = (z_1, z_2, \ldots, z_n)$ *— розв'язок системи лінійних заборон* $A \cdot x \neq \overline{0}$ *над полем* $\mathbb{F}_{2^k}$*. Тоді* $z' = (b \cdot z_1, b \cdot z_2, \ldots, b \cdot z_n)$*, де* $b \in \mathbb{F}_{2^k} \setminus \{0\}$ *— також розв'язок цієї системи.*

Доведення. Нехай z — розв'язок системи лінійних заборон, тоді

$$\begin{cases} a_1^{(1)} z_1 + a_2^{(1)} z_2 + \ldots + a_n^{(1)} z_n = y_1 \\ a_1^{(2)} z_1 + a_2^{(2)} z_2 + \ldots + a_n^{(2)} z_n = y_2 \\ \qquad\qquad \ldots \\ a_1^{(m)} z_1 + a_2^{(m)} z_2 + \ldots + a_n^{(m)} z_n = y_m \end{cases},$$

де $y_1, y_2, \ldots, y_m \in \mathbb{F}_{2^k} \setminus \{0\}$.

Помножимо ліву і праву частини усіх рівнянь на елемент поля $b \in \mathbb{F}_{2^k} \setminus \{0\}$.

$$\begin{cases} a_1^{(1)} z_1 b + a_2^{(1)} z_2 b + \ldots + a_n^{(1)} z_n b = y_1 b \\ a_1^{(2)} z_1 b + a_2^{(2)} z_2 b + \ldots + a_n^{(2)} z_n b = y_2 b \\ \qquad\qquad \ldots \\ a_1^{(m)} z_1 b + a_2^{(m)} z_2 b + \ldots + a_n^{(m)} z_n b = y_m b \end{cases}.$$

Виконаємо заміну $z'_i = z_i b$ для $i = \overline{1, n}$, $y'_j = y_j b$ для $j = \overline{1, m}$.

Отримуємо систему

$$\begin{cases} a_1^{(1)} z'_1 + a_2^{(1)} z'_2 + \ldots + a_n^{(1)} z'_n = y'_1 \\ a_1^{(2)} z'_1 + a_2^{(2)} z'_2 + \ldots + a_n^{(2)} z'_n = y'_2 \\ \ldots \\ a_1^{(m)} z'_1 + a_2^{(m)} z'_2 + \ldots + a_n^{(m)} z'_n = y'_m \end{cases}.$$

Оскільки $b \neq 0$, то нові $y'_j \neq 0$ для $j = \overline{1, m}$. Отже, $(z'_1, z'_2, \ldots, z'_n)$ — також розв'язок початкової системи лінійних заборон. □

Отже, за відомим розв'язком можна відновити частину інших розв'язків. Сформулюємо твердження, яке обчислює кількість таких розв'язків.

Твердження 2.5. *Якщо система лінійних заборон $A \cdot x \neq \overline{0}$ над полем $\mathbb{F}_{2^k}$ має хоча б один розв'язок, то потужність множини розв'язків принаймні $2^k - 1$.*

Доведення. Припустимо система лінійних заборон $A \cdot x \neq 0$ має розв'язок z'.

Визначимо операцію множення вектора на скаляр: $b \cdot z' = (b \cdot z'_1, b \cdot z'_2, \ldots, b \cdot z'_n)$ для $z \in \mathbb{F}_{2^k}^n$ та $b \in \mathbb{F}_{2^k}^*$. Розглянемо множину векторів

$$\{z \in \mathbb{F}_{2^k}^n | z = b \cdot z', b \in \mathbb{F}_{2^k}^*\}.$$

Покажемо, що для елементів $b_1 \neq b_2$, де $b_1, b_2 \in \mathbb{F}_{2^k}^*$, виконується $b_1 z' \neq b_2 z'$. Від супротивного: припустимо, що $b_1 z' = b_2 z'$. Це означає, що ці вектори збігаються у всіх компонентах, тобто: $b_1 z'_i = b_2 z'_i$ для $i = \overline{1, n}$. Очевидно, що розв'язок вихідної системи не може бути нульовим (оскільки в правих частинах заборон містяться нулі), тому виберемо номер j такий, що $z'_j \neq 0$. Тоді для компоненти з цим номером виконується $b_1 z'_j = b_2 z'_j$, а з цього випливає $(b_1 - b_2) z'_j = 0$. Оскільки $z'_j \neq 0$ та поле не має дільників нуля, то $b_1 = b_2$, тобто приходимо до суперечності. Таким чином, всі вектори $b \cdot z'$ для $b \in \mathbb{F}_{2^k}$ є різними.

Оскільки елемент b пробігає усю мультиплікативну групу поля $\mathbb{F}_{2^k}^*$,

потужність якої $2^k - 1$, то кількість векторів виду $b \cdot z'$ обмежена числом $2^k - 1$. Враховуючи, що всі вектори такого виду є різними, їхня кількість рівно $2^k - 1$. Отже, загальна кількість розв'язків системи лінійних заборон становить щонайменше $2^k - 1$. □

Сформулюємо теорему, яка описує структуру множини розв'язків системи лінійних заборон з нульовими правими частинами.

Теорема 2.2. *Нехай $D \subseteq \mathbb{F}_{2^k}^n$ — множина розв'язків системи лінійних заборон $A \cdot x \neq \overline{0}$ над полем $\mathbb{F}_{2^k}$, тоді $|D|$ ділиться на $2^k - 1$.*

Доведення. Розглянемо бінарне відношення на множині розв'язків D системи лінійних заборон: два вектори $a, b \in D$ знаходяться у бінарному відношенні $\sim$, якщо існує елемент $c \in \mathbb{F}_{2^k}^*$ такий, що $(a_1, a_2, \ldots, a_n) = (c \cdot b_1, c \cdot b_2, \ldots, c \cdot b_n)$. Таке відношення будемо називати *відношенням пропорційності*, а вектори, що належать цьому відношенню, — *пропорційними*.

Покажемо, що відношення $\sim$ є відношенням еквівалентності.

1) *Рефлексивність.* Властивість $\forall z \in D : z \sim z$ виконується зажди, оскільки можемо обрати $c = 1$.

2) *Симетричність.* Треба перевірити $\forall x, y \in D : x \sim y \Rightarrow y \sim x$. Якщо $x \sim y$, то існує $c \in \mathbb{F}_{2^k}^*$ такий, що $x_i = c \cdot y_i$ для $i = \overline{1, n}$. Оскільки $c \in \mathbb{F}_{2^k}^*$, то існує $c^{-1} \in \mathbb{F}_{2^k}^*$, тому $y_i = c^{-1} \cdot x_i$ для $i = \overline{1, n}$. Маємо $(y_1, y_2, \ldots, y_n) = (c^{-1} \cdot x_1, c^{-1} \cdot x_2, \ldots, c^{-1} \cdot x_n)$, а це означає, що $y \sim x$.

3) *Транзитивність.* Треба перевірити, що $\forall x, y, z \in D$ з $x \sim y, y \sim z$ випливає $x \sim z$. Припустимо, $x \sim y$ та $y \sim z$, тоді існують $c_1, c_2 \in \mathbb{F}_{2^k}^*$ такі, що $x_i = c_1 \cdot y_i$ та $y_i = c_2 \cdot z_i$ для $i = \overline{1, n}$. Підставляємо $y_i = c_2 \cdot z_i$ для $i = \overline{1, n}$ в умову $x \sim y$ і отримуємо $x_i = c_1 \cdot c_2 \cdot z_i$ для $i = \overline{1, n}$. Оскільки $c_1 \cdot c_2 \in \mathbb{F}_{2^k}^*$, то $x \sim z$.

Відношення еквівалентності на множині D породжує розбиття множини на класи еквівалентності: $D = D_1 \cup D_2 \cup \ldots D_s$, де $D_i \cap D_j = \varnothing$ при $i \neq j$, а s — кількість класів еквівалентності [34]. Таким чином, будь-які два елементи одного класу D_i, $i = \overline{1, s}$, знаходяться у відношенні

$\sim$, а будь-які два елементи різних класів D_i, D_j, $i \neq j$, не знаходяться у відношенні еквівалентності.

За твердженням 2.5 кількість елементів в одному класі еквівалентності становить $2^k - 1$. Оскільки всі класи не перетинаються, то

$$|D| = s \cdot (2^k - 1),$$

що і треба було довести. □

Зауваження. Кількість розв'язків однієї лінійної заборони становить $2^{kn} - 2^{k(n-1)} = 2^{k(n-1)}(2^k - 1)$, що узгоджується з доведеним фактом.

2.4 Системи лінійних заборон з нульовими правими частинами, що згенеровані невідомим фіксованим вектором

Припустимо, що вектор $z^{(tr)} \in \mathbb{F}_{2^k}^n$, де $n \geq 2$, є фіксованим. Будемо перебирати усі можливі вектори $a \in \mathbb{F}_{2^k}^n$ і всі ті, для яких виконується $(a, z^{(tr)}) \neq 0$, будемо записувати в множину A_{true}. Очевидно, що множина A_{true} буде складатись з попарно різних векторів.

Твердження 2.6. *Кількість векторів в множині* A_{true} *становить* $2^{kn} - 2^{k(n-1)}$.

Доведення. У твердженні 2.2 було доведено, що кількість розв'язків лінійної заборони складає $2^{kn} - 2^{k(n-1)}$. Розглянемо лінійну заборону $(a, x) \neq 0$, в якій x зафіксуємо значенням $z^{(tr)}$, а вектор a буде пробігати всі можливі значення $\mathbb{F}_{2^k}^n$. Тоді за доведеним твердженням кількість таких a для фіксованого x становить точно $2^{kn} - 2^{k(n-1)}$. □

Отже, $|A_{true}| = 2^{kn} - 2^{k(n-1)}$. Тепер сформуємо з усіх векторів цієї множини систему лінійних заборон і знайдемо її розв'язок D_{true}. Таким чином, побудували якомога повну систему лінійних заборон — додавати додаткові заборони немає сенсу, оскільки вичерпали всі можливі варіанти

заборон для фіксованого $z^{(tr)}$. Виникає питання чи можемо по повній системі заборони однозначно відновити розв'язок, тобто знайти той вектор $z^{(tr)}$, за допомогою якого була згенерована матриця A_{true}. Виявляється що так, тобто для системи $A \cdot x \neq \overline{0}$ можемо відновити розв'язок $z^{(tr)}$ з точністю до пропорційних йому векторів.

Теорема 2.3. *Для системи лінійних заборон $A_{true} \cdot x \neq 0$ кількість розв'язків $|D_{true}|$ дорівнює $2^k - 1$.*

Доведення. Нагадаємо, що $z^{(tr)}$ — це вектор, який використовувався при генеруванні A_{true}. Очевидно, що у системи лінійних заборон $A_{true} \cdot x \neq \overline{0}$ розв'язки існують, це випливає з побудови матриці A_{true}: на початку зафіксували значення $z^{(tr)}$ і додавали в A_{true} лише ті $a \in \mathbb{F}_{2^k}^n$, для яких $(a, z^{(tr)}) \neq 0$, тому жодний вектор, ортогональний $z^{(tr)}$, в A_{true} міститись не може, а тому $z^{(tr)}$ гарантовано є серед розв'язків.

За твердженням 2.5 маємо, що $|D_{true}| \geq 2^{k-1}$. Теорема 2.2 каже про структуру розв'язків системи: всі розв'язки розбиваються на класи еквівалентності за відношенням пропорційності, кожний з яких має потужність 2^{k-1}. Будемо доводити, що серед розв'язків системи лінійних заборон $A_{true} \cdot x \neq \overline{0}$ міститься лише один клас еквівалентності — це клас елементу $z^{(tr)}$.

На початку, зробимо декілька спостережень. Нехай i, де $1 \leq i \leq n$, — це позиція в векторі $z^{(tr)}$, на якій стоїть ненульовий елемент поля. Тоді заборона виду $a_i = (0, \ldots, 0, 1, 0, \ldots, 0)$, в якій на i-ій позиції стоїть одиниця, а на інших нулі, міститься в A_{true}, оскільки $(a_i, z^{(tr)}) = z_i^{(tr)} \neq 0$. Водночас $(a_i, z) = z_i$, тому z_i не може бути нулем. Таким чином, якщо на деякій позиції $z_i^{(tr)}$ стоїть ненульовий елемент поля, то на відповідній йому позиції вектора z_i теж має стояти ненульовий елемент.

Припустимо тепер, що i, де $1 \leq i \leq n$, — це позиція вектора $z^{(tr)}$, на якій стоїть нуль. Покажемо, що на відповідний позиції іншого довільного вектора z теж має стояти нуль. Будемо доводити від супротивного:

припустимо це не виконується, тобто $z_i \neq 0$. Виберемо будь-який інший елемент вектору $z^{(tr)}$, який не дорівнює нулю (такий існує, оскільки $z^{(tr)} \neq \overline{0}$), нехай він міститься на позиції j, де $1 \leq j \leq n$. Вже було доведено, що ненульовим елементам $z^{(tr)}$ відповідають ненульові елементи z, тому $z_j \neq 0$. Побудуємо вектор b таким чином: $b_i = z_i^{-1}$, $b_j = z_j^{-1}$, а на всіх інших позиціях містяться нулі. Тоді $(b, z^{(tr)}) = z_j^{(tr)} z_j^{-1} \neq 0$, а $(b, z) = z_i \cdot z_i^{-1} + z_j \cdot z_j^{-1} = 0$. Отже, z не є розв'язком, а це суперечить припущенню. Тому нульовим позиціям $z^{(tr)}$ відповідають нульові позиції z. Отже, в множину кандидатів на розв'язок можуть попасти лише ті вектори, у яких ненульові позиції точно збігаються з $z^{(tr)}$. Надалі мова буде йти тільки про вектори такого виду.

Фактично, потрібно довести, що якого б кандидата не взяли (звісно, він має бути не пропорційним по відношенню до $z^{(tr)}$), в множині A_{true} завжди знайдеться заборона, яка буде відкидати цього кандидата. Більш формально, $\forall z$, який не є пропорційним до $z^{(tr)}$, існує $a \in A_{true}$ такий, що $(a, z) = 0$.

Розглянемо більш детально умову пропорційності. По-перше, звузимо коло кандидатів, то тих, у яких щонайменше дві ненульові компоненти, оскільки якщо компонента одна (тоді, як відомо, у векторі $z^{(tr)}$ буде єдина ненульова компонента на тій самій позиції), то вектор є пропорційним до $z^{(tr)}$. Також будемо розглядати тільки позиції з ненульовими компонентами, оскільки у випадку, коли одна компонента з двох буде нулем, а інша ні, умова пропорційності не буде порушуватись. З огляду на ці зауваження, будемо казати, що вектор z не є пропорційним до $z^{(tr)}$, якщо існує пара індексів i, j, де $1 \leq i, j \leq n$ та $i \neq j$, таких, що $z_i z_j^{(tr)} + z_i^{(tr)} z_j = d$, де $d \neq 0$, причому $z_i, z_i^{(tr)}, z_j, z_j^{(tr)} \neq 0$. Якби ж на цих двох позиціях умова пропорційності не порушувалась, то мали б місце рівності $z_i = c \cdot z_i^{(tr)}$, $z_j = c \cdot z_j^{(tr)}$, звідки $c = z_i \cdot (z_i^{(tr)})^{-1} = z_j \cdot (z_j^{(tr)})^{-1}$, тому $z_i \cdot (z_i^{(tr)})^{-1} = z_j \cdot (z_j^{(tr)})^{-1}$, а з цього випливає $z_i z_j^{(tr)} + z_i^{(tr)} z_j = 0$. Роблячи підсумок, необхідно показати, що для кандидатів z таких, що $\exists\ i \neq j,\ z_i z_j^{(tr)} + z_i^{(tr)} z_j = d \neq 0$, існує заборона $a \in A_{true}$ така, що

$(a, z) = 0$, тобто z не є розв'язком.

Зафіксуємо вектор-кандидат z, який задовольняє описаним вище умовам. Нехай a — це вектор, який складається з нулів, окрім позицій i та j, на яких порушується пропорційність векторів z та $z^{(tr)}$. Побудуємо вектор a, задавши компоненти a_i та a_j:

$$a_i = d \cdot z_i^{-1} + z_j^{(tr)}, \quad a_j = z_i^{(tr)}.$$

Нагадаємо, що $z_i, z_j, z_i^{(tr)}, z_j^{(tr)} \neq 0$ та

$$d = z_i z_j^{(tr)} + z_i^{(tr)} z_j \neq 0.$$

Перевіримо чи належить a множині A_{true}, тобто $(a, z^{(tr)}) \neq 0$:

$$a_i z_i^{(tr)} + a_j z_j^{(tr)} = (d \cdot z_i^{-1} + z_j^{(tr)}) z_i^{(tr)} + z_i^{(tr)} z_j^{(tr)} = d \cdot z_i^{-1} \cdot z_i^{(tr)}.$$

Оскільки $d, z_i^{-1}, z_i^{(tr)} \neq 0$, то цей вектор міститься в A_{true}.

Перевіримо чи виконується $(a, z) = 0$:

$$a_i z_i + a_j z_j = (d \cdot z_i^{-1} + z_j^{(tr)}) z_i + z_i^{(tr)} z_j = d + z_i z_j^{(tr)} + z_i^{(tr)} z_j = d + d = 0.$$

Ці перевірки завершують доведення.

□

Доведена теорема каже про те, що точно відновити $z^{(tr)}$ не можливо, але можна знайти множину пропорційних векторів, серед яких гарантовано буде міститись $z^{(tr)}$, якщо маємо якомога повну систему заборон. Але виникає проблема, яка полягає в тому, що розмір $|A_{true}|$ занадто великий, наприклад, для $\mathbb{F}_8$ та $n = 5$, кількість всіх векторів становить 32768, а $|A_{true}| = 28672$, тобто складає 87.5%. В загальному випадку,

$$\frac{2^{kn} - 2^{k(n-1)}}{2^{kn}} = 1 - \frac{2^{kn-k}}{2^{kn}} = 1 - 2^{-k},$$

і при $k \to \infty$ прямує до одиниці. З практичної точки зори, це унеможливлює розв'язок системи лінійних заборон, оскільки потребує,

щоб вона містила дуже велику кількість векторів.

Це приводить до питання, як по заданим параметрам задачі оцінити мінімальну кількість векторів, які мають міститись в системі, щоб отримати в результаті $|\widetilde{D}|$ розв'язків, де $|\widetilde{D}|$ приблизно дорівнює $|D_{true}|$ (у випадку нульових правих частин $|D_{true}|$ становить $2^k - 1$).

З точки зору практики це може знадобитись в такій ситуації: нехай є необмежений доступ до векторів $a \in \mathbb{F}_{2^k}^n$, таких що $(a, z^{(tr)}) \neq 0$ (наприклад, можемо звертатись до оракула, який на кожному кроці випадково надає один з таких векторів). В такому разі є певна прив'язка до кількості запитів до оракула (або часу), тому необхідно розуміти скільки мінімально треба назбирати таких попарно різних векторів (і чи є різниця які саме вектори обирати, а якщо є, то яка стратегія вибору є оптимальною з точки зору найшвидшого зменшення кількості розв'язків), щоб отримати систему лінійних заборон з кількістю розв'язків, яка є якомога ближчою до $|D_{true}| = 2^k - 1$.

Формалізуємо це питання таким означенням.

Означення 2.5. *Точкою насичення* будемо називати число $S = \min_{A'} |A'|$, де мінімум обчислюється по всім таким матрицям A', що $A' \subseteq A_{true}$ та $A' \cdot x \neq 0$ має $|D_{true}|$ розв'язків. Відповідно, матрицю, на якій цей мінімум досягається, будемо називати *насиченою матрицею*.

Фактично, насичена матриця — це така матриця, що передає усі властивості A_{true}, але при цьому може мати набагато менший розмір. Якщо знати точку насичення, то це надасть уяву скільки запитів до оракула треба зробити, перед тим як починати розв'язувати систему лінійних заборон. Якщо почати розв'язувати систему з ненасиченою матрицею, то може виникнути ситуація, коли потужність множини розв'язків буде занадто великою. Проілюструємо на прикладі, що значення точки насичення на практиці є набагато меншим за розмір A_{true}.

Приклад 2.1. Дано поле $\mathbb{F}_8$ та фіксований вектор $z^{(tr)}$ розміру $n = 5$. Тоді нескладно встановити розмір матриці $|A_{true}|$ та множини розв'язків

$|D_{true}|$: $|A_{true}| = 28672$ та $|D_{true}| = 7$. Тепер спробуємо знайти залежність кількості розв'язків в системі від кількості заборон.

На кожній ітерації виконуємо таку послідовність дій.

1) Формуємо матрицю A_{true}.

2) Ініціалізуємо $D = \mathbb{F}_{2^k}^n$.

3) Рівноймовірно та незалежно обираємо вектор a з множини A_{true} (вибірка здійснюється з поверненням).

4) Знаходимо розв'язок заборони $\left(a, z^{(tr)}\right) \neq 0$, який позначаємо D_j.

5) Знаходимо перетин $D := D \cap D_j$ — кількість розв'язків системи лінійних заборон на поточному кроці.

Тепер можемо побудувати графік залежності $|D|$ від кількості ітерацій. Цей графік зображено на рисунку 2.1.

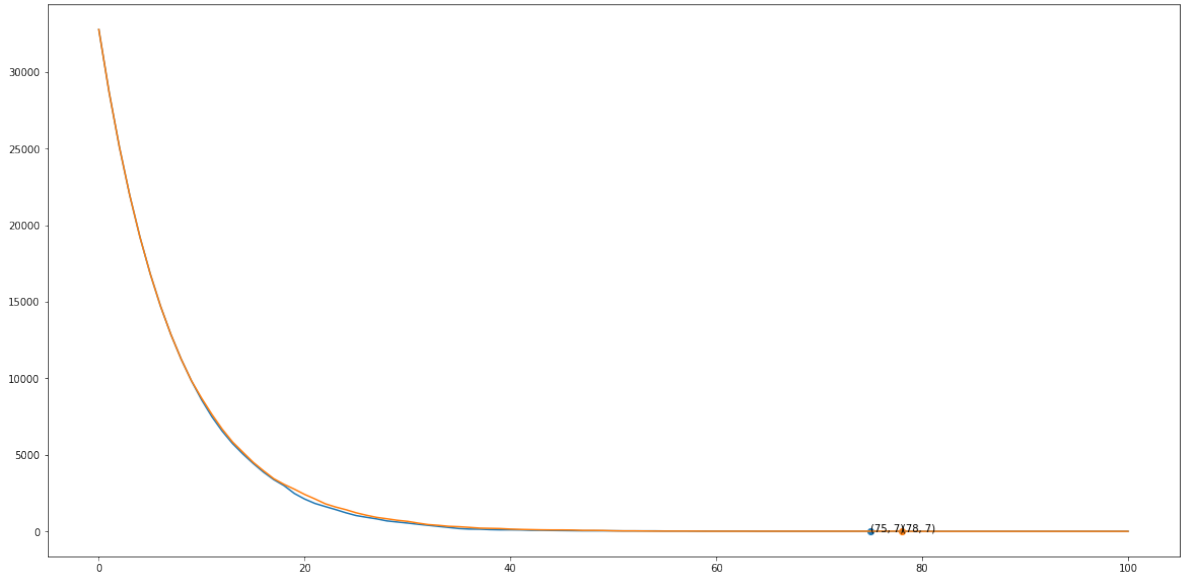

Рисунок 2.1 – Залежність кількості розв'язків від кількості рівнянь в системі лінійних заборон

Бачимо, що точка насичення приймає значення $70-80$, що є набагато меншим за $|A_{true}| = 28672$.

Зауваження. Якби мали справу з системою лінійних рівнянь, то потужність перетину D спадала б набагато швидше, оскільки кількість розв'язків лінійного рівняння становить $2^{k(n-1)}$, що є незначною кількістю

по відношенню до 2^{kn}, тому що $2^{k(n-1)}$ поділити на 2^{kn} дорівнює 2^{-k}. В цьому полягає відмінність систем лінійних заборон від систем лінійних рівнянь з комбінаторної точки зору.

Зауважимо, що в ситуації, коли невідомий вектор x є фіксованим, у системи лінійних заборон завжди будуть існувати розв'язки, тобто питання про існування розв'язків не виникає. Ситуація, в якій система $A \cdot x \neq \overline{0}$ не має розв'язків, можлива лише в тому випадку, коли деякі заборони в системі були породжені іншими невідомими векторами, і такі заборони заперечують допустимі значення одна одної.

2.5 Системи лінійних заборон з ненульовими правими частинами, що згенеровані невідомим фіксованим вектором

Системи лінійних заборон з ненульовими правими частинами зручно задавати за допомогою множини виду $\widehat{A} \subset \{\langle a, a_0\rangle | a \in \mathbb{F}_{2^k}^n, a_0 \in \mathbb{F}_{2^k}\}$, де кожна пара $\langle a, a_0\rangle$ задає лінійну заборону $(a, x) \neq a_0$. Також систему лінійних заборон з ненульовими правими частинами можна розглядати як матрицю розміру $(n+1) \times m$, в якій останній стовпчик складається з елементів $a_0 \in \mathbb{F}_{2^k}$. Таким чином, в залежності від контексту будемо розглядати систему заборон з ненульовими правими частинами як матрицю з $n+1$ стовпчиками або як множину пар $\langle a, a_0\rangle$ — всі перелічені об'єкти вважатимуться еквівалентними способами представлення системи лінійних заборон з ненульовими правими частинами.

Припустимо, що праві частини системи лінійних заборон можуть містити не тільки нульовий елемент поля, а й інші. Як і раніше, вектор $z^{(tr)} \in \mathbb{F}_{2^k}^n$ є фіксованим. Будемо розглядати нетривіальний випадок $z^{(tr)} \neq 0$. Для довільного елемента b поля $\mathbb{F}_{2^k}$ визначимо множину

$$A_{true}^{(b)} = \{\langle a, b\rangle, a \in \mathbb{F}_{2^k}^n | (a, z^{(tr)}) \neq b\}.$$

Зауважимо, що з точністю до першої компоненти кожної пари $A_{true}^{(0)}$

збігається з A_{true}. Виберемо довільний ненульовий елемент поля g і сформуємо для нього множину $A^{(g)}_{true}$. Побудуємо множину

$$\widehat{A}_{true} = A^{(0)}_{true} \cup A^{(g)}_{true},$$

яка буде задавати систему лінійних заборон з ненульовими правими частинами. Через $\widehat{D}_{true}$ будемо позначати кількість розв'язків такої системи лінійних заборон.

Теорема 2.4. *Для системи лінійних заборон $\widehat{A}_{true}$ кількість розв'язків $|\widehat{D}_{true}| = 1$.*

Доведення. Нагадаємо, що $z^{(tr)}$ — це фіксований вектор, яким було згенеровано множину $\widehat{A}_{true}$. Очевидно, що $z^{(tr)}$ є розв'язком такої системи, оскільки в цій системі за побудовою не міститься жодного вектора, який би забороняв $z^{(tr)}$.

Необхідно довести, що для будь-якого вектору $z \in \mathbb{F}^n_{2^k}$, $z \neq z^{(tr)}$, існує пара $\langle a, b\rangle \in \widehat{A}_{true}$ така, що $(a, z) = b$. Тобто який би вектор z, за винятком справжнього розв'язку, не обрали, в системі завжди знайдеться заборона, яка буде відкидати z.

Оскільки $A^{(0)}_{true} \subseteq \widehat{A}_{true}$, то можемо застосувати міркування з доведення теореми 2.3, в якому стверджувалось, що довільний вектор z має ненульові елементи поля на тих самих позиціях, що і вектор $z^{(tr)}$, інакше z не є розв'язком.

Покажемо тепер як відкинути всі інші вектори. Нехай на i-ій позиції, $1 \leq i \leq n$, векторів $z^{(tr)}$ та z містяться ненульові елементи поля. Покажемо, що з $z_i \neq z^{(tr)}_i$ випливає, що z не є розв'язком. Побудуємо заборону a таким чином: на i-ій позиції буде елемент $g \cdot z_i^{-1}$, а на всіх інших нулі. Тоді $(a, z^{(tr)}) = g \cdot z_i^{-1} \cdot z^{(tr)}_i$. Цей скалярний добуток буде дорівнювати g тільки в тому разі, коли $z_i^{-1} \cdot z^{(tr)}_i = 1$, тобто $z_i = z^{(tr)}_i$, а це не виконується за припущенням, отже $(a, z^{(tr)}) \neq g$. У випадку деякого іншого вектору z маємо: $(a, z) = g \cdot z_i^{-1} \cdot z_i = g$, отже z не може бути розв'язком. □

З оглядом на цей факт, можемо зробити висновок, що при наявності достатньої кількості лінійних заборон з нульовими та ненульовими частинами, можемо точно відновити $z^{(tr)}$, який було згенеровано $\widehat{A}_{true}$. Означення точки насичення та насиченої матриці для множини $\widehat{A}_{true}$ є аналогічним означенню 2.5. В контексті системи з ненульовими правими частинам також постає питання пошуку матриць з малою кількістю заборон, але які однозначно задають невідомий фіксований розв'язок.

Доведення теореми 2.4 є конструктивним і надає можливість будувати матриці з ненульовими правими частинами, які містять невелику кількість рівнянь і мають єдиний розв'язок.

Твердження 2.7. *Нехай $\widehat{A}_{expl} \subseteq \widehat{A}_{true}$ — це система лінійних заборон, яка складається з усіх можливих заборон одного з двох типів.*

1) Заборони, в яких одна компонента дорівнює 1, всі інші компоненти дорівнюють нулю та права частина також дорівнює нулю.

2) Заборони, в яких одна компонента дорівнює деякому ненульовому елементу поля, всі інші компоненти дорівнюють нулю, а права частина дорівнює фіксованому ненульовому елементу поля g.

Тоді така система має єдиний розв'язок $z^{(tr)}$.

Доведення. Розглянемо які вектори забороняє кожна з описаних в умові груп заборон.

Розглянемо лінійні заборони виду $\langle a_i, 0\rangle$, де $a_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0, 0)$ та 1 міститься на позиції з номером $1 \leq i \leq n$. Припустимо, що $z_i^{(tr)} \neq 0$, тоді заборона $\langle a_i, 0\rangle$ міститься в системі $\widehat{A}_{expl}$. Тому для деякого іншого вектора z виконується $(a_i, z) = z_i$, звідки $z_i \neq 0$. Отже, з $z_i^{(tr)} \neq 0$ випливає $z_i \neq 0$.

Розглянемо лінійні заборони виду $\langle b_i^{(d)}, g\rangle$, де $b_i^{(d)} = (0, 0, \ldots, 0, d, 0, \ldots, 0, 0)$ та довільний ненульовий елемент поля d міститься на позиції з номером $1 \leq i \leq n$. Якщо $z_i^{(tr)} \neq 0$ (а тоді і $z_i \neq 0$ — це було доведено раніше), то для фіксованого номера i в систему

попадуть всі заборони з $d \neq (z_i^{(tr)})^{-1}g$, оскільки при $d = (z_i^{(tr)})^{-1}$ маємо

$$(b_i^{(d)}, z_i^{(tr)}) = d \cdot z_i^{(tr)} = (z_i^{(tr)})^{-1} \cdot g \cdot z_i^{(tr)} = g.$$

Отже, в системі містяться заборони виду $(b_i^{(d)}, x) \neq g$, де $d \in \mathbb{F}_{2^k}^* \setminus \{(z_i^{(tr)})^{-1}g\}$. Тоді для деякого іншого вектору z розглянемо, при яких значеннях компоненти $z_i \in \mathbb{F}_{2^k}^*$ вектор z не буде розв'язком. Для цього обчислимо $(b_i^{(d)}, z_i) = d \cdot z_i$, тоді має виконуватись $d \cdot z_i = g$, звідки $z_i = d^{-1} \cdot g$. Оскільки d приймає усі значення з мультиплікативної групи поля, окрім значення $(z_i^{(tr)})^{-1}g$, то вектор z не може бути розв'язком для всіх таких значень d, за виключенням $d = (z_i^{(tr)})^{-1}g$. Таким чином, відкинули усі можливі значення z_i, окрім

$$z_i = d^{-1} \cdot g = z_i^{(tr)} g^{-1} \cdot g = z_i^{(tr)}.$$

Отже, з $z_i^{(tr)} \neq 0$ випливає $z_i = z_i^{(tr)}$, тобто всі ненульові компоненти вектора $z^{(tr)}$ збігаються з усіма ненульовими елементами вектора z.

Нехай для фіксованого індексу i, де $1 \leq i \leq n$, компонент $z_i^{(tr)} = 0$, тоді в систему лінійних заборон попадуть усі заборони $\langle b_i^{(d)}, g \rangle$ з ненульовим d на позиції i. Тому для будь-якого іншого вектору z виконується $(b_i^{(d)}, z) = d \cdot z_i \neq g$. Покажемо, що такі заборони виключать усі можливі вектори, в яких $z_i \in \mathbb{F}_{2^k}^*$. Оскільки d пробігає усі елементи $\mathbb{F}_{2^k}^*$, то $z_i \neq g \cdot d^{-1}$, де $\{g \cdot d^{-1} | d \in \mathbb{F}_{2^k}^*\} = \mathbb{F}_{2^k}^*$, тому залишається єдиний варіант $z_i = 0$. Отже, з $z_i^{(tr)} = 0$ випливає $z_i = 0$.

Таким чином, для будь-якого $z \in \mathbb{F}_{2^k}^n$ маємо $z = z^{(tr)}$. □

Зауваження. Доведене твердження виконується для випадку $z^{(tr)} = \overline{0}$.

Зауваження. Вибір g в означенні матриць $\widehat{A}_{true}$ та $\widehat{A}_{expl}$ є несуттєвим, головне щоб g не дорівнював нулю. Можна обрати, наприклад, $g = 1$, оскільки всі результати з побудовою і аналізом матриці $\widehat{A}_{expl}$ не залежать від вибору g, а потребують лише $g \in \mathbb{F}_{2^k}^*$. Але будемо наводити всі твердження для довільного $g \in \mathbb{F}_{2^k}^*$, вважаючи, що він

завжди фіксований деякою константою.

Виявляється, можна точно обчислити кількість векторів у множині $\widehat{A}_{expl}$.

Твердження 2.8. *Кількість векторів в матриці* $\widehat{A}_{expl}$ *становить* $(2^k - 1) \cdot n$.

Доведення. Будемо користуватись введеними у твердженні 2.7 позначеннями a_j та $b_j^{(d)}$, де $1 \leq j \leq n$, $d \in \mathbb{F}_{2^k}^*$.

Нехай $1 \leq i \leq n$ — фіксована ненульова позиція вектору $z^{(tr)}$. Тоді рівно одна заборона $\langle a_i, 0 \rangle$ потрапить в $\widehat{A}_{expl}$. Кількість заборон типу $\langle b_i^{(d)}, g \rangle$ буде становити $2^k - 2$, оскільки тільки у разі $d = (z_i^{(tr)})^{-1} g$ буде виконано $(b_i^{(d)}, z^{(tr)}) = g$, тобто порушиться умова приналежності матриці $\widehat{A}_{expl}$. Іншим виключенням є $d = 0$. Отже, одна ненульова компонента додає $2^k - 2 + 1 = 2^k - 1$ заборон в $\widehat{A}_{expl}$.

Нехай тепер $1 \leq i \leq n$ — фіксована нульова позиція вектору $z^{(tr)}$. Тоді жодна заборона $\langle a_i, 0 \rangle$ не потрапить в матрицю $\widehat{A}_{expl}$, оскільки скалярний добуток $(a_i, z^{(tr)})$ буде дорівнювати нулю. Кількість заборон $\langle b_i^{(d)}, g \rangle$, які потраплять в матрицю, буде становити $2^k - 1$, оскільки яка б i-та координата вектору $b_i^{(d)}$ не була, скалярний добуток $(b_i^{(d)}, z^{(tr)})$ буде дорівнювати нулю, а фіксований g гарантовано не є нулем. Залишається лише зауважити, що в означенні вектору $b_i^{(d)}$ на i-ій компоненті стояв ненульовий елемент поля d, тому кількість таких векторів буде дорівнювати $2^k - 1 + 0 = 2^k - 1$. Отже, нульова компонента також додає $2^k - 1$ заборон в $\widehat{A}_{expl}$.

Загальна кількість заборон в множині $\widehat{A}_{expl}$ буде дорівнювати $n \cdot (2^k - 1)$. Зауважимо також, що кількість лінійних заборон виду $\langle a_i, 0 \rangle$, де $1 \leq i \leq n$, збігається з кількістю ненульових елементів вектора $z^{(tr)}$, тому за всіма можливими заборонами такого виду можна точно відтворити усі ненульові позиції вектору $z^{(tr)}$. □

Твердження 2.9. *Для систем лінійних заборон з фіксованим розв'язком та заборонами, в яких праві частини дорівнюють нулю або*

фіксованому $g \neq 0$*, точка насичення* $S \leq (2^k - 1) \cdot n$*.*

Доведення. За твердженням 2.7 матриця $\widehat{A}_{expl}$ має рівно один розв'язок та є підмножиною $\widehat{A}_{true}$, тому ця матрицю є насиченою за означенням. За твердженням 2.8 така матриця містить $(2^k - 1) \cdot n$ різних векторів, а оскільки в означенні точки насичення мінімум знаходиться по всім насиченим матрицям, то величина $(2^k - 1) \cdot n$ буде верхньою оцінкою для точки насичення. □

Розглянемо таку множину лінійних заборон з ненульовими правими частинами:

$$\widehat{A}'_{true} = \bigcup_{g \in \mathbb{F}_{2^k}} A^{(g)}_{true}.$$

Ця множина є якомога повною множиною заборон для фіксованого вектору $z^{(tr)}$.

Твердження 2.10. *Нехай* $\widehat{A}'_{expl} \subset \widehat{A}'_{true}$ *— це система лінійних заборон, яка складається з усіх можливих заборон одного з двох типів.*

1) Заборони, в яких одна компонента дорівнює 1*, всі інші компоненти дорівнюють нулю та права частина також дорівнює нулю.*

2) Заборони, в яких одна компонента дорівнює фіксованому ненульовому елементу поля r*, всі інші компоненти дорівнюють нулю, а права частина дорівнює довільному ненульовому елементу поля* s*.*
Тоді така система має єдиний розв'язок $z^{(tr)}$*.*

Доведення. Покажемо, що побудована таким чином система лінійних заборон є еквівалентною системі $\widehat{A}_{expl}$ з твердження 2.7 для фіксованого вектору $z^{(tr)}$, відносно яких ці системи були згенеровані. Фіксовану праву частини $\widehat{A}_{expl}$ будемо позначати g. Побудуємо бієктивне відображення між множинами заборон $\widehat{A}'_{expl}$ та $\widehat{A}_{expl}$, а також покажемо, що таке відображення для кожної з заборон зберігає множину розв'язків.

Будемо користуватись введеними у твердженні 2.7 позначеннями a_j та $b_j^{(r)}$, де $1 \leq j \leq n$, $r \in \mathbb{F}^*_{2^k}$.

Побудуємо відображення $f : \widehat{A}'_{expl} \to \widehat{A}_{expl}$ таким чином.

1) Якщо заборона y має вид $\langle a_i, 0\rangle$, то $f(y)$ просто повертає y.

2) Якщо заборона y має вид $\langle b_i^{(r)}, s\rangle$, де $s, r \neq 0$ та r — фіксований елемент $\widehat{A}'_{expl}$, то $f(y)$ повертає заборону $\langle b_i^{(rs^{-1}g)}, g\rangle$, де g — фіксований елемент $\widehat{A}_{expl}$.

Оскільки $\widehat{A}'_{expl}$ та $\widehat{A}_{expl}$ згенеровані одним і тим самим фіксованим вектором $z^{(tr)}$, то для кожної заборони з $\widehat{A}'_{expl}$ завжди буде існувати заборона з $\widehat{A}_{expl}$ відповідно до заданого відображення.

Перевіримо ін'єктивність та сюр'єктивність.

1) *Ін'єктивність.* Покажемо, що з $y_1 \neq y_2$ випливає $f(y_1) \neq f(y_2)$. Для заборон виду $\langle a_i, 0\rangle$ це слідує з ін'єктивності тотожної функції. Покажемо для заборон другого типу. Нехай для фіксованої позиції $1 \leq i \leq n$ заборони $y_1 = \langle b_i^{(r)}, s_1\rangle$, $y_2 = \langle b_i^{(r)}, s_2\rangle$, $s_1 \neq s_2$, тоді $f(y_1) = \langle b_i^{(gs_1^{-1}r)}, g\rangle$, $f(y_1) = \langle b_i^{(gs_2^{-1}r)}, g\rangle$, а збігатись вони можуть тоді і тільки тоді, коли $gs_1^{-1}r = gs_2^{-1}r$, тобто коли $s_1 = s_2$, а це суперечить припущенню.

2) *Сюр'єктивність.* Покажемо, що для кожної заборони $y \in \widehat{A}_{expl}$ існує розв'язок рівняння $f(x) = y$. Для заборон виду $\langle a_i, 0\rangle$ це випливає з ін'єктивності тотожної функції. Покажемо для заборон другого типу. Нехай $y = \langle b_i^{(d)}, g\rangle$ для деякого $d \neq 0$. Тоді можемо покласти $x = \langle b_i^{(r)}, g \cdot d^{-1}r\rangle$ та обчислити значення функції f від цього аргументу:

$$f(x) = \langle b_i^{(r\cdot g^{-1}dr^{-1}\cdot g)}, g\rangle = \langle b_i^{(d)}, g\rangle.$$

Таким чином, побудували бієкцію між множинами $\widehat{A}'_{expl}$ та $\widehat{A}_{expl}$. З цього випливає, що кількість заборон в цих системах є однаковою.

Тепер покажемо, що кожна заборона $x \in \widehat{A}'_{expl}$ буде еквівалентна забороні $f(x) \in \widehat{A}'_{expl}$, тобто кількість їх розв'язків збігається. Для заборон типу $\langle a_i, 0\rangle$ це очевидно виконується, тому будемо доводити лише для заборон другого типу.

Нехай z є розв'язком заборони $\langle b_i^{(r)}, s\rangle \in \widehat{A}'_{expl}$ для деякого $s \neq 0$ та фіксованого $1 \leq i \leq n$. Тоді для цього вектору z виконується $(b_i^{(r)}, z) \neq s$,

тобто $r \cdot z_i \neq s$, звідки $z_i \neq s \cdot r^{-1}$. Відповідна їй заборона з $\widehat{A}_{expl}$ буде мати вид $\langle b_i^{(rs^{-1}g)}, g\rangle$. Покажемо, що будь-який z, для якого $z_i \neq s \cdot r^{-1}$, буде також розв'язком заборони $\langle b_i^{(rs^{-1}g)}, g\rangle$. Помножаючи обидві частини заборони $z_i \neq s \cdot r^{-1}$ на $rs^{-1}g$ маємо: $rs^{-1}g \cdot z_i \neq g$, звідки випливає, що такий z є розв'язком заборони $\langle b_i^{(rs^{-1}g)}, g\rangle$.

Аналогічно можна показати, що будь-який розв'язок заборони $\langle b_i^{(d)}, g\rangle \in \widehat{A}_{expl}$ для фіксованих $d \neq 0$ та позиції $1 \leq i \leq n$ є розв'язком заборони $\langle b_i^{(r)}, g \cdot d^{-1}r\rangle \in \widehat{A}'_{expl}$. Отже, множини розв'язків відповідних лінійних заборон збігаються, а тому збігаються і множини розв'язків систем лінійних заборон $\widehat{A}_{expl}$ та $\widehat{A}'_{expl}$.

□

Припустимо є джерело випадкових заборон O_r, яке в кожний момент часу генерує заборону $\langle a, a_0\rangle$, де $a \in \mathbb{F}_{2^k}^n$ та $a_0 \in \mathbb{F}_{2^k}$. Припустимо також, що не можна побудувати усю матрицю $\widehat{A}'_{true}$, тому що наявні обмежені за часом ресурси. Виникає питання, чи можемо в такому разі робити певні локальні припущення про вектор $z^{(tr)}$. В цьому випадку має місце таке спостереження: якщо O_r ще не згенерувало заборону $\langle a_i, 0\rangle$ для деякого $1 \leq i \leq n$, то не можна точно відрізнити дві ситуації — заборона $\langle a_i, 0\rangle \notin \widehat{A}'_{true}$ або O_r ще не встигло її згенерувати. По цій самій причині не можемо відновити жодну з матриць $\widehat{A}_{expl}$ або $\widehat{A}'_{expl}$. Виникає питання чи можна в такому разі отримувати деяку інформацію про $z^{(tr)}$.

Твердження 2.11. *Нехай $b_i^{(d)} = (0, \ldots, 0, d, 0, \ldots, 0)$, де $d \in \mathbb{F}_{2^k}^*$ міститься на позиції з номером $1 \leq i \leq n$. Тоді виконуються такі твердження.*

1) Якщо O_r згенерувало заборону $\langle b_i^{(d)}, 0\rangle$, то i-та компонента вектора $z^{(tr)} \neq 0$.

2) Якщо O_r згенерувало вектор $\langle b_i^{(d)}, g\rangle$, де $g \neq 0$, то i-та компонента вектора $z^{(tr)}$ або дорівнює нулю або не дорівнює нулю та елементу $g \cdot d^{-1}$ одночасно.

Доведення. Розглянемо два випадки.

1) Якщо O_r згенерувало $\langle b_i^{(d)}, 0\rangle$, то $(b_i^{(d)}, z^{(tr)}) = d \cdot z_i^{(tr)} \neq 0$. Від супротивного: припустимо, що $z_i^{(tr)} = 0$, тоді $(b_i^{(d)}, z^{(tr)}) = 0$, тобто маємо суперечність.

2) Якщо O_r згенерувало вектор $\langle b_i^{(d)}, g\rangle$, то $(b_i^{(d)}, z^{(tr)}) = d \cdot z_i^{(tr)} \neq g$. Якщо $z_i^{(tr)} = 0$, то $d \cdot z_i^{(tr)} = 0 \neq g$. Якщо $z_i^{(tr)} \neq 0$, то $d \cdot z_i^{(tr)} \neq g$, звідки випливає $z_i^{(tr)} \neq g \cdot d^{-1}$.

□

Таким чином, якщо заборони мають певний вигляд, то можна робити припущення про невідомий вектор. З цього випливає, що системи лінійних заборон, в яких велика кількість заборон з однією ненульовою компонентою в лівій частині, надають змогу оптимізувати перебір всіх можливих варіантів.

2.6 Пошук алгоритму розв'язку систем лінійних заборон над скінченним полем

Перейдемо до спроб побудувати алгоритм для розв'язку системи лінійних заборон. Найпростіший спосіб — це перебирати усі можливі вектори розміру n над полем $\mathbb{F}_{2^k}$ та для кожного з векторів перевіряти чи є цей вектор розв'язком.

Алгоритм 2.2. *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$ над полем $\mathbb{F}_{2^k}$.

Для кожного вектору $x_0 \in \mathbb{F}_{2^k}^n$:

1) Покласти $ind = 1$, $D = \varnothing$.

2) Для кожного $j = \overline{1, m}$:

а) Перевірити чи виконується $(a^{(j)}, x_0) \neq 0$.

б) Якщо не виконується, то покласти $ind = 0$ та перервати внутрішній цикл.

3) Якщо $ind = 1$, то додати x_0 до множини D.

Вихід. Множина розв'язків D.

Складність методу повного перебору становить $m \cdot n \cdot 2^{nk}$ операцій

множення в полі. Будемо враховувати лише операції множення в полі, оскільки операція XOR потребує набагато менше обчислювальних ресурсів та нею можна знехтувати.

При дослідженні математичного об'єкта одним з основних інструментів є зведення до наявних задач або спроба застосувати наявні методи. Системи лінійних заборон за своєю структурою схожі на системи лінійних нерівностей, системи рівнянь зі спотвореними правими частинами та системі лінійних рівнянь. Виникає ідея спробувати перенести алгоритми розв'язку для перелічених об'єктів на системи лінійних заборон. Наведемо перелік деяких методів та аргументуємо, чому вони не застосовні до систем заборон або не надають прискорення у порівнянні з методом повного перебору.

1) Система лінійних нерівностей складається з виразів виду $(a, x) < a_0$ та $(a, x) > a_0$. Виникає ідея замінити заборону на пару відношень «<» та «>», тобто перетворити заборону типу $(a, x) \neq 0$ на $(a, x) < 0$ та $(a, x) > 0$. Тоді б отримали $2 \cdot m$ нерівностей і можна було б застосувати до них відомі методи, але відношення «<» та «>» не мають сенсу для скінченного поля, оскільки на елементах поля не визначено відношення порядку. Тому з такими лінійними заборонами не можна оперувати як з нерівностями у випадку поля $\mathbb{F}_{2^k}$.

2) Виконаємо таке перетворення над вхідною системою — додамо до кожного рівняння штучні змінні, які будемо вважати випадковим шумом, та замінимо усі знаки «$\neq$» на знаки «=»:

$$\begin{cases} a_1^{(1)}x_1 + a_2^{(1)}x_2 + \ldots + a_n^{(1)}x_n + \varepsilon_1 = 0 \\ a_1^{(2)}x_1 + a_2^{(2)}x_2 + \ldots + a_n^{(2)}x_n + \varepsilon_2 = 0 \\ \ldots \\ a_1^{(m)}x_1 + a_2^{(m)}x_2 + \ldots + a_n^{(m)}x_n + \varepsilon_m = 0 \end{cases},$$

де $\varepsilon_j \neq 0$ для $j = \overline{1, m}$. Отримана система нагадує систему зі спотвореними правими частинами, тому виникає гіпотеза застосувати

відповідні методи. Проблема полягає в тому, що розв'язок систем зі спотвореними правими частинами сильно залежить від припущень щодо розподілу шуму, а в даному випадку не можемо висунути ніяких (навіть суто практичних) гіпотез щодо розподілу шуму, тому можемо вважати що він є рівноймовірним на всій множині елементів поля $\mathbb{F}_{2^k}$. В такому разі оцінка максимальної правдоподібності не зможе знайти найімовірніших кандидатів у розв'язок серед векторів розміру n над полем $\mathbb{F}_{2^k}$.

3) Для зручності перенумеруємо додаткові змінні, які були введені у минулому пункті: $\varepsilon_{n+1}, \varepsilon_{n+2}, \ldots, \varepsilon_{n+m}$. Матриця, яка відповідає новій системі, має розмір $m \times (n+m)$:

$$\widehat{A} = \begin{pmatrix} a_1^{(1)} & a_2^{(1)} & \ldots & a_n^{(1)} & 1 & 0 & \ldots & 0 \\ a_1^{(2)} & a_2^{(2)} & \ldots & a_n^{(2)} & 0 & 1 & \ldots & 0 \\ & & \ldots & & & & & \\ a_1^{(m)} & a_2^{(m)} & \ldots & a_n^{(m)} & 0 & 0 & \ldots & 1 \end{pmatrix}.$$

Можемо застосувати до цієї матриці метод Гауса, щоб звести її до виду трапеції:

$$\widehat{A} = \begin{pmatrix} 1 & b_2^{(1)} & \ldots & b_n^{(1)} & b_{n+1}^{(1)} & \ldots & b_m^{(1)} & b_{m+1}^{(1)} & \ldots & b_{n+m}^{(1)} \\ 0 & 1 & \ldots & b_n^{(2)} & b_{n+1}^{(2)} & \ldots & b_m^{(2)} & b_{m+1}^{(2)} & \ldots & b_{n+m}^{(2)} \\ & & & & & \ldots & & & & \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 1 & b_{m+1}^{(m)} & \ldots & b_{n+m}^{(m)} \end{pmatrix}.$$

Якщо матриця приведена до виду трапеції, то ранг матриці дорівнює кількості ненульових рядків, тому $\operatorname{rank} \widehat{A} = m$. Кількість змінних в системі дорівнює $n+m$, тому система буде мати безліч розв'язків. Для цього щоб описати всі ці розв'язки необхідно обрати базис системи

розв'язків. Якщо останній рядок матриці $\hat{A}$ записати у вигляді рівняння, то він буде мати такий вигляд:

$$\varepsilon_m + b_{m+1}^{(n)}\varepsilon_{m+1} + \ldots + b_{n+m-1}^{(n-1)}\varepsilon_{n+m-1} + b_{n+m}^{(n)}\varepsilon_{n+m} = 0.$$

За базис можна обрати множину $\{\varepsilon_{m+1}, \ldots, \varepsilon_{n+m}\}$, розмір якої становить n. Тепер можемо записати ε_m у цьому базисі та підставити його у $(m-1)$ рівняння системи. Таким чином, система вже не буде включати ε_m. При повторенні цієї процедуру на деякому кроці отримаємо вираз для x_n

$$x_n + b_{n+1}^{(n)}\varepsilon_{n+1} + \ldots + b_m^{(n)}\varepsilon_m + b_{m+1}^{(n)}\varepsilon_{m+1} + \ldots + b_{m+n}^{(n)}\varepsilon_{n+m} = 0.$$

Аналогічно виражаємо усі $x_1, \ldots, x_{n-1}$. Отже, через цей базис виражаються $m-n$ штучних змінних $\varepsilon_{n+1}, \varepsilon_{n+2}, \ldots, \varepsilon_m$ та n початкових змінних $x_1, x_2, \ldots, x_n$. Залишається підставити деякий набір значень $(\varepsilon_{m+1}, \ldots, \varepsilon_{n+m})$ та обчислити невідомі змінні, але в цьому випадку принциповим є те, що всі ε_i для $i = \overline{(n+1), m}$ не можуть дорівнювати нулю. Таким чином, приходимо до задачі, дуже подібної до початкової: необхідно розв'язати систему лінійних рівнянь відносно ε, враховуючи заборону на ε, але ця система має простіший вигляд за рахунок приведення до форми трапеції. Виходить, що застосувавши метод Гауса, отримуємо незначну оптимізацію при переборі значень $(\varepsilon_{m+1}, \ldots, \varepsilon_{n+m})$, оскільки отримана система буде містити більше нулів, але початкова складність задачі при цьому зберігається. З цього можемо зробити висновок, що типовими лінійними методами систему лінійних заборон не можна розв'язати.

З огляду на розглянуті спроби, робимо висновок, що потрібно шукати перетворення іншого типу. Щобільше, це перетворення має задовольняти такій властивості: при введенні нових змінних чи будь-яких інших операціях зі змінними не потрібно явно накладати якісь обмеження на ці змінні (такими обмеженнями можуть виступати лінійні заборони). У випадку, коли в рамках перетворення необхідно накладати заборони,

отримуємо початкову задачу, представлену в іншому вигляді, тобто її обчислювальна складність зберігається.

Альтернативним перетворенням може бути так званий «трюк Рабіновича». Полягає він в наступному: нехай нам потрібно заборону $x \neq 0$, для $x \in \mathbb{F}_{2^k}$ замінити на рівність. Можемо ввести нову змінну $\gamma \in \mathbb{F}_{2^k}$ і замінити дану заборону на рівняння $1 + \gamma \cdot x = 0$. Покажемо, що при такій заміні множина розв'язків початкової системи заборон залишається незмінною з точністю до введених штучних змінних.

Твердження 2.12. *Множина розв'язків лінійної системи заборон виду*

$$\begin{cases} (a^{(1)}, x) \neq 0 \\ (a^{(2)}, x) \neq 0 \\ \qquad \dots \\ (a^{(m)}, x) \neq 0 \end{cases},$$

збігається з множиною розв'язків лінійної системи рівнянь виду

$$\begin{cases} 1 + \gamma_1 \cdot (a^{(1)}, x) = 0 \\ 1 + \gamma_2 \cdot (a^{(2)}, x) = 0 \\ \qquad\qquad \dots \\ 1 + \gamma_m \cdot (a^{(m)}, x) = 0 \end{cases},$$

з точністю до штучних змінних $\gamma_1, \gamma_2, \dots, \gamma_m$, *де* $\gamma_j \in \mathbb{F}_{2^k}$ *для* $j = \overline{1, m}$.

Доведення. Нехай $\widetilde{D}$ — це множина розв'язків системи заборон, яка складається з векторів розміру n, а R — множина розв'язків системи рівнянь, яка складається з векторів розміру $n+m$. Домовимось, що кожний вектор з множини R спочатку містить $x_1, x_2, \dots, x_n$, а потім $\gamma_1, \gamma_2, \dots, \gamma_m$.

Будемо позначати $x_{1:n}$ — перші n компоненти вектору x.

Покажемо, що якщо $x \in \widetilde{D}$, то існує $y \in R$ такий, що $y_{1:n} = x$. Якщо x є розв'язком системи лінійних заборон, то для нього існує такий набір

елементів $z_1, z_2, \ldots, z_m$, $z_j \neq 0$ для $j = \overline{1,m}$, що виконуються рівності

$$\begin{cases} (a^{(1)}, x) = z_1 \\ (a^{(2)}, x) = z_2 \\ \qquad \ldots \\ (a^{(m)}, x) = z_m \end{cases}.$$

Підставляємо всі z_i , $i = \overline{1,m}$, у систему рівнянь та маємо:

$$\begin{cases} \gamma_1 \cdot z_1 = 1 \\ \gamma_2 \cdot z_2 = 1 \\ \qquad \ldots \\ \gamma_m \cdot z_m = 1 \end{cases}.$$

Покладемо $\gamma_j = (z_j)^{-1}$ для $j = \overline{1,m}$ і отримуємо розв'язок y, який належить множині R, і в якому перші n компонент збігаються з вектором x.

Покажемо, що якщо $y \in R$, то $y_{1:n} \in \widetilde{D}$. Якщо y є розв'язком системи рівнянь, то

$$\begin{cases} 1 + y_{n+1} \cdot (a^{(1)}, y_{1:n}) = 0 \\ 1 + y_{n+2} \cdot (a^{(2)}, y_{1:n}) = 0 \\ \qquad \ldots \\ 1 + y_m \cdot (a^{(n)}, y_{1:n}) = 0 \end{cases}.$$

Звідси випливає, що $(a^{(j)}, y_{1:n})$ не дорівнює нулю для $j = \overline{1,m}$, оскільки у разі, коли для деякого номеру $1 \leq i \leq m$ вираз $(a^{(i)}, y_{1:n})$ перетворюється в нуль, маємо суперечність $1 = 0$. А це своєю чергою й означає, що $y_{1:n} \in \widetilde{D}$.

Отже, $\widetilde{D}$ та R збігаються з точністю до n перших компонент вектору.

□

Зауваження. При такому перетворенні кількість розв'язків в множині R збігається з кількістю розв'язків в множині $\widetilde{D}$, тобто $|R| = |\widetilde{D}|$. Це забезпечує, що у множині R не містяться копії деякого

початкового розв'язку з різними m останніми компонентами вектору.

Зауважимо, що наявність вільних змінних у правих частинах ніяк не впливає на хід доведення. Отже, отримали таку систему квадратичних рівнянь над полем $\mathbb{F}_{2^k}$:

$$\begin{cases} 1 + a_0^{(1)}\gamma_1 + a_1^{(1)}x_1\gamma_1 + a_2^{(1)}x_2\gamma_1 + \ldots + a_n^{(1)}x_n\gamma_1 = 0 \\ 1 + a_0^{(2)}\gamma_2 + a_1^{(2)}x_1\gamma_2 + a_2^{(2)}x_2\gamma_2 + \ldots + a_n^{(n)}x_n\gamma_2 = 0 \\ \ldots \\ 1 + a_0^{(m)}\gamma_m + a_1^{(m)}x_1\gamma_m + a_2^{(m)}x_2\gamma_m + \ldots + a_n^{(m)}x_n\gamma_m = 0 \end{cases}.$$

Отримана система є частковим випадком задачі MQ. Відомо, що задача MQ є $\mathcal{NP}$-повною задачею. Запропонований підхід пов'язує системи лінійних заборон з системами квадратичних рівнянь, отже до систем лінійних заборон можуть бути використані методи дослідження систем квадратичних рівнянь. Такі методи розглянуті в підрозділі 1.1.

Введемо такі позначення:

$$\Gamma = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_m) = \begin{pmatrix} \gamma_1 & 0 & \ldots & 0 \\ 0 & \gamma_2 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & \gamma_m \end{pmatrix}, \quad \widehat{a}_0 = \overline{1} + a_0 = \begin{pmatrix} 1 + a_0^{(1)} \\ 1 + a_0^{(2)} \\ \ldots \\ 1 + a_0^{(m)} \end{pmatrix},$$

тоді систему можна записати в такому вигляді:

$$\Gamma \cdot A \cdot x = \widehat{a}_0.$$

Зауважимо, що в такому представленні принциповим є те, що діагональна матриця Γ міститься в лівій частині системи. Для того, щоб відділити Γ від A, необхідно помножити обидві частини цієї рівності на Γ^{-1}. Згідно з критерієм існування оберненої матриці має виконуватись $\det\Gamma \neq 0$. У випадку діагональної матриці $\det\Gamma = \prod_{i=1}^{m}\gamma_i$. Якщо

параметризувати добуток γ_i, $i = \overline{1,m}$, деяким ненульовим елементом g, тобто покласти $\prod_{i=1}^{m} \gamma_i = g \in \mathbb{F}_{2^k}^*$, то можна переписати систему у вигляді $A \cdot x \neq \Gamma^{-1} \cdot \widehat{a}_0$. Таким чином, отримуємо ще одне перетворення, подібне до введення ненульового шуму в систему, яке має декілька недоліків. По-перше, у системі з'являється рівняння $\prod_{i=1}^{m} \gamma_i = g$ степеню m, по-друге, таку систему необхідно розглядати для всіх можливих параметрів g, тобто замість однієї вихідної системи необхідно розглядати набір з $2^k - 1$ систем. З цього можна зробити висновок, що матрицю Γ доцільно не відділяти від лівої частини системи.

Висновки до розділу 2

Сформульовано та доведено критерій існування розв'язку системи лінійних заборон. Цей критерій зводить задачу перевірки існування розв'язків системи лінійних заборон до задачі перевірки полінома на тотожну рівність нулю над скінченним полем. Оскільки поліном задано у вигляді добутку лінійних функцій, то складність такого зведення є поліноміальною. На основі цього критерію побудовано поліноміальний імовірнісний алгоритм перевірки існування розв'язку для випадку $m \leq 2^{k-1}$. Проведено аналіз цього алгоритму та обчислено імовірність його односторонньої помилки, яка дорівнює 2^{-d}, де d — це деяке фіксоване число, яке визначає порядок точності алгоритму. Цей алгоритм було імплементовано та перевірено на деякому наборі вхідних параметрів.

Доведено властивості розв'язку систем лінійних заборон у випадку нульових правих частин. Ці властивості характеризують структуру множини розв'язків системи лінійних заборон та надають змогу знаходити додаткові розв'язки у ситуації, коли хоча б один розв'язок є відомим.

Розглянуто випадок, коли система лінійних заборон згенерована невідомим фіксованим вектором. Такий випадок виникає у практичних застосуваннях, оскільки невідомим фіксованим вектором може бути певний параметр, який необхідно відновити (наприклад, ключ шифрування). Досліджено питання принципової можливості відновити невідомий вектор за системою лінійних заборон, яка містить максимально можливу кількість заборон. Оскільки систему лінійних заборон, яка містить максимально можливу кількість заборон, складно побудувати на практиці, то було сформульовано задачу побудови системи лінійних заборон, яка містить відносно невелику кількість лінійних заборон, але надає змогу однозначно відновити невідомий вектор. Запропоновано один з можливих способів розв'язання цієї задачі за допомогою побудови системи спеціального виду, яка містить $n \cdot (2^k - 1)$ заборон та відкидає усі вектори розміру n, окрім того вектора, яким цю систему було згенеровано.

Оскільки система лінійних заборон за структурою своїх лівих частин є подібною до системи лінійних рівнянь та системи лінійних рівнянь зі спотвореними правими частинами, то була проаналізована можливість застосування наявних методів розв'язку систем лінійних рівнянь зі звичайними та спотвореними правими частинами до системи лінійних заборон. Серед усіх розглянутих методів було виділено «трюк Рабіновича», який надає змогу перетворити систему лінійних заборон у систему квадратичних рівнянь, яка є частковим випадком задачі MQ. Тому для перетвореної таким чином системи можна застосовувати наявні алгоритми розв'язку задачі MQ.

3 ДОСЛІДЖЕННЯ СКЛАДНОСТІ ЗАДАЧ, ПОВ'ЯЗАНИХ ІЗ СИСТЕМАМИ ЛІНІЙНИХ ЗАБОРОН

З точки зору теорії складності, пошук невідомого розв'язку системи лінійних заборон є обчисленням, яке потребує певних ресурсів. В практичних застосуваннях обов'язковою умовою є ефективність такого обчислення, оскільки в іншому разі для роботи алгоритму можуть знадобитись занадто великі ресурси. Саме тому важливим етапом аналізу математичного об'єкта є оцінка складності задач, які пов'язані з цим об'єктом. Оскільки різні обчислювальні пристрої можуть мати різну архітектуру, то необхідно абстрагуватись від конкретного пристрою та вимірювати складність задачі у кількості певних елементарних операцій. Тому універсальним способом оцінки складності задачі є доведення приналежності цієї задачі певному класу складності. В цьому розділі для деяких задач, пов'язаних з системами лінійних заборон, визначаються їхні класи складності та досліджуються властивості цих задач з обчислювальної точки зору.

3.1 Основні задачі, пов'язані із системами лінійних заборон

Сформулюємо задачу існування розв'язку довільної системи лінійних заборон над скінченним полем.

Задача 3.1 (SLR-decision)**.** *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$ над полем $\mathbb{F}_{2^k}$, вектор a_0 розміру $m \times 1$.

Необхідно з'ясувати чи існує розв'язок над полем $\mathbb{F}_{2^k}$ у системи лінійних заборон $Ax \neq a_0$.

Вихід. «Так», якщо розв'язок існує, «Ні», інакше.

Зауважимо, що в формулюванні задачі на вхід подається все поле $\mathbb{F}_{2^k}$. Це необхідно для того, щоб при ітерації всіх елементів поля,

представлених у вигляді масиву з бітових рядків, залишитись в рамках поліноміального обчислення. Якби на вхід подавався тільки степінь розширення поля, то цикл по всім елементам поля вважався би не ефективним обчисленням. Таке кодування вхідних даних не є занадто надлишковим, оскільки перебір всіх можливих векторів фіксованої довжини над скінченним полем залишається експоненційним обчисленням.

Найбільш універсальний спосіб оцінки складності задачі — це доведення приналежності цієї задачі деякому класу складності. Для задачі SLR-decision можна побудувати поліноміальний алгоритм перевірки розв'язку, тому вона належить класу складності $\mathcal{NP}$. Елементарними операціями в подальших твердженнях будемо вважати операції добутку елементів в полі. Операціями додавання в полі можна знехтувати.

Твердження 3.1. *Задача SLR-decision належить класу задач* $\mathcal{NP}$.

Доведення. Для того, щоб довести приналежність класу $\mathcal{NP}$, покажемо для SLR-decision існує сертифікат, довжина якого обмежена поліномом від довжини входу, а також те, що процедура перевірки сертифікату є поліноміальною за часом від довжини входу задачі.

Існування поліноміального сертифіката. Сертифікатом для цієї задачі є розв'язок системи лінійних заборон $(x_1, x_2, \ldots, x_n)$. Довжина цього вектору становить n, що, очевидно, обмежено поліномом від розміру вхідних даних.

Поліноміальна перевірка сертифікату. Перевірка сертифікату полягає фактично в підставленні $(x_1, x_2, \ldots, x_n)$ в систему лінійних заборон і перевірці чи задовольняє цей вектор усім заборонам. Процедуру перевірки можна формалізувати таким алгоритмом.

Вхід. Вектор $(x_1, x_2, \ldots, x_n)$.

1) Для кожного $j = \overline{1, m}$:

   а) Перевірити чи виконується $a_1^{(j)} x_1 + a_2^{(j)} x_2 + \ldots + a_n^{(j)} x_n \neq a_0^{(j)}$.

   б) Якщо не виконується, то повернути «Ні».

2) Повернути «Так».

Вихід. «Так», якщо $(x_1, x_2, \ldots, x_n)$ є розв'язком системи лінійних заборон, «Ні» інакше.

Процедура перевірки потребує $n \cdot m$ операцій множення в полі $\mathbb{F}_{2^k}$, отже є поліноміальною від довжини вхідних даних задачі. □

Нагадаємо, що критерій 2.1 зводив питання існування розв'язку до перевірки полінома на тотожну рівність нулю, але не надавав можливість знаходити сам розв'язок. Оскільки на практиці часто необхідно знаходити сам розв'язок, то сформулюємо задачу знаходження розв'язку системи лінійних заборон.

Задача 3.2 (SLR-search)**.** *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$ над полем $\mathbb{F}_{2^k}$, вектор a_0 розміру $m \times 1$.

Необхідно знайти хоча б один розв'язок над полем $\mathbb{F}_{2^k}$ у системи лінійних заборон $Ax \neq a_0$.

Вихід. Розв'язок системи лінійних заборон $Ax \neq a_0$ або «$\perp$», якщо його не існує.

Розглянемо твердження, яке, в певному сенсі, зводить задачу пошуку розв'язку по задачі перевірки існування розв'язку.

Твердження 3.2. *Задачі SLR-decision та SLR-search є еквівалентними відносно звідності по Тюрінгу:*

$$\textit{SLR-decision} =_T \textit{SLR-search}.$$

Доведення. Для того, щоб довести еквівалентність по Тюрінгу двох задач, необхідно і достатньо показати, що кожна з цих задач є звідною по Тюрінгу до іншої.

SLR-decision $\leq_T$ *SLR-search.* Щоб довести цю звідність, необхідно показати, що існує поліноміальний алгоритм B, який за допомогою поліноміальної кількості звернень до оракула, що розв'язує задачу SLR-search, розв'язує задачу SLR-decision.

В цьому алгоритм B будується таким чином: отримуючи на вхід

$(\mathbb{F}_{2^k}, A, a_0)$, B звертається до оракула SLR-decision, отримує x — розв'язок системи лінійних заборон, якщо він існує; інакше отримує «$\perp$». Відповідно, якщо він отримав x, то повертає відповідь «Так», інакше повертає «Ні».

SLR-search $\leq_T$ *SLR-decision.* Щоб довести цю звідність, необхідно показати, що існує поліноміальний алгоритм B, який за допомогою поліноміальної кількості звернень до оракула, що розв'язує задачу SLR-decision, розв'язує задачу SLR-search.

Ідея полягає в тому, щоб за допомогою звернень до оракула SLR-search, покомпонентно відновлювати шуканий вектор $x = (x_1, x_2, \ldots, x_n)$. Для відновлення першої компоненти x_1 вектору x переносимо доданок $a_1^{(j)} x_1$ у праву частину для всіх $j = \overline{1, m}$ (за твердженням 2.3 це не змінює множину розв'язків системи) та послідовно фіксуємо значення x_1 елементами поля $\mathbb{F}_{2^k}$. Якщо на деякому елементі d отримали відповідь «Так», то фіксуємо компоненту x_1 цим значенням d та модифікуємо початкову систему лінійних заборон, замінюючи праві частини $a_0^{(j)}$ на $a_0^{(j)} + a_1^{(j)} d$ для $j = \overline{1, m}$. Переходимо до пошуку наступної компоненти з модифікованими матрицею A' розмірності $m \times (n-1)$ та вектором a'.

З обчислювальної точки зору для відновлення однієї компоненти необхідно щонайбільше $|\mathbb{F}_{2^k}| = 2^k$ операцій звернення до оракула. Для всіх компонент вектору необхідно $n \cdot 2^k$ операцій звернення до оракула, що є поліноміальною кількістю від довжини входу (оскільки на вхід отримуємо поле $\mathbb{F}_{2^k}$ та значення n).

Коректність алгоритму випливає з того, що при пошуку кожної наступної координати фіксуємо лише те значення, при якому існує розв'язок всієї системи. Процес пошуку вектору x можна представити у вигляді дерева, де кожна вершина має $|\mathbb{F}_{2^k}|$ нащадків, а кількість листів дерева складає 2^{kn}. Задача пошуку полягає в знаходженні шляху до листу дерева, що відповідає значенню вектора x, який є розв'язком системи лінійних заборон, а оракул, фактично, допомагає запобігти

перебору всіх можливих шляхів, тобто шукати шлях рівень за рівнем, відкидаючи на кожному рівні експоненційну кількість помилкових варіантів. Оскільки на кожному кроці обираємо саме те значення компоненти вектора, при якому увесь вектор з урахуванням невідомих компонент є розв'язком, то гарантовано знаходимо розв'язок. Таким чином, розв'язок не відновиться тільки в тому випадку, якщо на деякому кроці обрали гілку зі значенням компоненти вектору x, на яке було отримано відповідь оракула «Ні». Якщо система розв'язків не має, то на першій же компоненті отримаємо від оракулу всі відповіді «Ні». □

3.2 Алгоритми пошуку розв'язку систем лінійних заборон

Для отримання поліноміального імовірнісного алгоритму пошуку розв'язку у випадку $m \leq 2^{k-1}$ алгоритм 2.1 можна скомбінувати з результатами теореми 3.2.

Теорема 3.2 описує спосіб відновлення вектору-розв'язку системи лінійних заборон за допомогою $n \cdot 2^k$ звернень до оракула. Замість звернення до оракула будемо використовувати алгоритм 2.1. Він дозволяє з урахуванням умови $m \leq 2^{k-1}$ розв'язувати задачу пошуку розв'язку систем лінійних заборон з односторонньою помилкою 2^{-d}, де d — це фіксована точність, яка встановлюється заздалегідь. Складність цього алгоритму становить $d \cdot m \cdot n$ операцій множення в полі. Будемо позначати цей алгоритм Sol_exists.

Алгоритм 3.1. *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$ та вектор a_0 розміру $m \times 1$, де $m \leq 2^{k-1}$.

Відсортувати елементи поля $\mathbb{F}_{2^k}$ в поліноміальному базисі як бітові рядки за порядком зростання. Ці бітові рядки будуть формувати масив B, де $|B| = 2^k$.

Зафіксувати $d \in \mathbb{N}$, тоді помилка одного запуску Sol_exists буде 2^{-d}.

Цикл по i від 1 до n:

1) Цикл по j від 1 до 2^k:

а) Сформувати матрицю A' з матриці A, видаливши в лівих частинах усіх заборон доданок з x_i. Сформувати вектор a'_0 з вектора a_0, присвоївши $a'^{(j)}_0$ значення $a^{(j)}_0 + a^{(j)}_i B_j$ для $j = \overline{1, m}$.

б) Обчислити $d = Sol\_exists(\mathbb{F}_{2^k}, A', a'_0)$.

в) Якщо $d =$ «Так», то покласти $x_i = B_j$, $A = A'$, $a_0 = a'_0$ та перервати цикл по j.

2) Якщо $d =$ «Ні», то повернути «$\perp$».

Повернути вектор $(x_1, x_2, \ldots, x_n)$.

Вихід. Вектор $(x_1, x_2, \ldots, x_n)$ або «$\perp$».

Зауваження. Наведений алгоритм пошуку розв'язку можна спростити, якщо алгоритм 2.1, окрім відповіді «Так», буде ще повертати вектор, на якому виконується $F(x) \neq 0$. Тоді для пошуку розв'язку буде необхідно лише n перевірок для першої компоненти. Але в такому разі алгоритм 3.1 втратить свою універсальність. Тому цей алгоритм побудовано так, щоб замість алгоритму 2.1 можна було використовувати довільний поліноміальний алгоритм перевірки існування розв'язку з односторонньою помилкою 2^{-d}.

Отже, загальна складність алгоритму в найгіршому випадку складає $O(n^2 \cdot m \cdot 2^k \cdot d)$ і залишається поліноміальною від довжини входу. Оскільки значення d, яке фіксує порядок помилки, встановлюється заздалегідь, то можна обрати його набагато більшим за кількість ітерацій імовірнісного алгоритму, щоб запобігти накопиченню помилки в рамках усього алгоритму. В такому разі помилка не буде накопичуватись, оскільки кількість звертань до Sol_exists є поліноміальною, а функція 2^{-d} зменшується експоненційно зі збільшенням d.

Зауважимо, що такий алгоритм можна застосовувати і у випадку $m > 2^{k-1}$, але тоді він стає емпіричним, оскільки в такому разі відсутнє теоретичне обґрунтування коректності його роботи. Можливість його використання спричинена тим, що алгоритм 2.1 перевірки існування розв'язків має односторонню помилку, тобто він може випадково

відкидати деякі дійсні розв'язки, але ніколи не повертає вектори, які не є розв'язками. Таким чином може бути ситуація, коли розв'язок системи лінійних заборон існував, а алгоритм пошуку розв'язку повернув «$\perp$», але не може бути ситуації, коли цей алгорим повернув вектор, який не є розв'язком.

Розглянемо емпіричний алгоритм, який дозволяє шукати декілька розв'язків системи лінійних заборон. Під $Find\_one\_sol(\mathbb{F}_{2^k}, A, a_0)$ будемо позначати поліноміальний імовірнісний алгоритм 3.1, який знаходить один розв'язок системи лінійних заборон з високою імовірністю. Можна застосувати алгоритм знаходження одного розв'язку таким чином: знайшовши деякий розв'язок, накласти на нього заборону, та продовжити ітеративно запускати імовірнісний алгоритм для пошуку наступних розв'язків, додаючи для кожного з них відповідні заборони в систему. В цьому випадку, час роботи алгоритму вже не обмежено поліномом, оскільки кількість розв'язків системи лінійних заборон потенційно може бути експоненційною, тобто не обмеженою поліномом від довжини вхідних даних. Для випадку нульових правих частин теорема 2.2 надає представлення про структуру розв'язків, а точніше вона стверджує, що кількість розв'язків становить $s \cdot (2^k - 1)$, де s — кількість класів еквівалентності за відношенням пропорційності. Таким чином, для випадку нульових правих частин після знаходження одного розв'язку необхідно додати в множину розв'язків його клас еквівалентності, тобто множину пропорційних йому векторів.

Щоб реалізувати такий алгоритм, необхідно вміти для знайденого розв'язку шукати заборону. Такою забороною може бути вектор, ортогональний знайденому розв'язку. Задача пошуку ортогонального вектора не є складною, можна скористатись детермінованим алгоритмом 3.2.

Алгоритм 3.2 ($Orth\_search$)**.** *Вхід.* Вектор $x \in \mathbb{F}_{2^k}^n$.

1) Якщо $x = \overline{0}$, то повернути вектор $u = (1, 0, \ldots, 0)$.

2) Якщо $x \neq \overline{0}$, то існує принаймні одна ненульова компонента.

Будемо позначати позицію цієї компоненти i, де $1 \leq i \leq n$.

3) Покласти $I = \{1, 2, \ldots, n\}$ — множина індексів вектору x. Обчислити

$$s = \sum_{j \in I \setminus \{i\}} x_j.$$

4) Повернути вектор u такого виду:

$$u = (1, 1, \ldots, 1, x_i^{-1} \cdot s, 1, \ldots, 1),$$

де $x_i^{-1} \cdot s$ міститься на позиції з індексом i.

Вихід. Вектор $u \neq \overline{0}$ такий, що $(u, x) = 0$.

Переконаємось, що $u \neq \overline{0}$ справді є ортогональним до x:

$$(u, x) = 1 \cdot x_1 + \ldots + x_i^{-1} s \cdot x_i + \ldots + 1 \cdot x_n = s + s = 0.$$

Зауважимо, що запропонований спосіб побудови ортогонального вектора не єдиний. Нехай $1 \leq i \leq n$ — це індекс ненульової компоненти вектора x, тоді можна випадково згенерувати $n - 1$ елемент поля $u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n$ та обчислити

$$s = \sum_{j \in I \setminus \{i\}} r_j u_j.$$

Тоді якщо $u_i = x_i^{-1} \cdot s$, то отриманий вектор u теж буде ортогональним вектору x.

Маючи процедуру пошуку ортогонального вектора, можемо сформулювати алгоритм 3.3 пошуку декількох розв'язків системи лінійних заборон.

Алгоритм 3.3. *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$, вектор a_0 розміру $m \times 1$, де $m \leq 2^{k-1}$.

Ініціалізувати $D = \varnothing$, $A' = A$.

Повторювати:

1) $d = Find\_one\_sol(\mathbb{F}_{2^k}, A', a_0)$.

2) Якщо d не дорівнює «$\perp$», то:

а) Якщо $a_0 = \overline{0}$, то $D_0 = \{c \cdot d, c \in \mathbb{F}_{2^k}^*\}$, інакше $D_0 = \{d\}$.

б) Оновити $D = D \cup D_0$.

в) Знайти вектор $u = Orth\_search(d)$.

г) Оновити матрицю A', додавши туди вектор u. Оновити вектор a_0, додавши туди елемент 0.

Поки d не дорівнює «$\perp$».

Вихід. Підмножина D множини розв'язків системи лінійних заборон $A \cdot x \neq a_0$.

Цей алгоритм є евристичним, оскільки коли додаємо заборону на кожному кроці, то відкидаємо не гарантовано 1 розв'язок (або $2^k - 1$ розв'язків у випадку $a_0 = \overline{0}$), а щонайменше 1 (або $2^k - 1$ у випадку $a_0 = \overline{0}$), — може трапитись, що в множині розв'язків знайшовся розв'язок, ортогональний забороні, яку додали до системи лінійних заборон. Під час практичного застосування цього алгоритму було емпірично встановлено, що найкращі результати він показує, коли потужність множини розв'язків є невеликою. Наприклад, у випадку поля $\mathbb{F}_8$ та $n = 5$ при кількості розв'язків приблизно $2 - 10$, алгоритм знаходив всі ці розв'язки. Отже, цей алгоритм доцільно застосовувати в ситуації, коли система лінійних заборон має достатню кількість лінійних заборон та її множина розв'язків складається з невеликої кількості векторів.

3.3 Складність часткових випадків задачі перевірки існування розв'язку систем лінійних заборон

Оскільки наразі невідома більш точна оцінка складності задачі SLR-decision, аніж приналежність класу $\mathcal{NP}$, то сформулюємо часткові випадки цієї задачі та оцінимо їхню складність.

Розглянемо версію задачі SLR-decision, в якій поле є фіксованим і дорівнює $\mathbb{F}_2$.

Задача 3.3 ($\mathbb{F}_2$-SLR-decision)**.** *Вхід.* Матриця A розміру $m \times n$ над полем $\mathbb{F}_2$, вектор a_0 розміру $m \times 1$ над полем $\mathbb{F}_2$.

Необхідно знайти хоча б один розв'язок над полем $\mathbb{F}_2$ системи лінійних заборон $Ax \neq a_0$.

Вихід. «Так», якщо розв'язок існує, «Ні», інакше.

Розглянемо твердження, яке стосується складності задачі $\mathbb{F}_2$-SLR-decision. Зауважимо, що за своєю структурою ця задача подібна задачі XORSAT.

Твердження 3.3. *Задача $\mathbb{F}_2$-SLR-decision належить класу складності $\mathcal{P}$.*

Доведення. Якщо поле в умові задачі є фіксованим і дорівнює $\mathbb{F}_2$, то для системи лінійних заборон

$$\begin{cases} a_1^{(1)}x_1 + a_2^{(1)}x_2 + \ldots + a_n^{(1)}x_n \neq y_1 \\ a_1^{(2)}x_1 + a_2^{(2)}x_2 + \ldots + a_n^{(2)}x_n \neq y_2 \\ \qquad\qquad \ldots \\ a_1^{(m)}x_1 + a_2^{(m)}x_2 + \ldots + a_n^{(m)}x_n \neq y_m \end{cases},$$

де $a_i^{(j)}, y_j \in \mathbb{F}_2$ для $i = \overline{1,n}, j = \overline{1,m}$, можна отримати еквівалентну їй систему рівнянь, виконавши заміну $\widetilde{y_j} = y_j + 1$ для $j = \overline{1,m}$:

$$\begin{cases} a_1^{(1)}x_1 + a_2^{(1)}x_2 + \ldots + a_n^{(1)}x_n = \widetilde{y_1} \\ a_1^{(2)}x_1 + a_2^{(2)}x_2 + \ldots + a_n^{(2)}x_n = \widetilde{y_2} \\ \qquad\qquad \ldots \\ a_1^{(m)}x_1 + a_2^{(m)}x_2 + \ldots + a_n^{(m)}x_n = \widetilde{y_m} \end{cases}.$$

Для розв'язку таких систем існують поліноміальні алгоритми, наприклад, метод Гауса. Він може бути використаний для обчислення рангу матриці, яка задає систему лінійних рівнянь, а маючи ранг матриці, можна надати відповідь на питання чи існують розв'язки в вихідній системі лінійних рівнянь. Складність алгоритму Гауса є кубічною відносно довжини

вхідних даних, тому вона обмежена поліномом від довжини вхідних даних. □

Розглянемо апроксимаційну версію задачі SLR-decision, в якій не обов'язково знаходити розв'язок для всіх лінійних заборон в системі, а можна знайти частковий розв'язок, який задовольняє щонайменше $1 \leq l \leq m$ лінійних заборон.

Задача 3.4 (Max-SLR-decision)**.** *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$ над полем $\mathbb{F}_{2^k}$, вектор a_0 розміру $m \times 1$, число l, де $1 \leq l \leq m$.

Необхідно з'ясувати чи існує розв'язок над полем $\mathbb{F}_{2^k}$ у системи, що утворена щонайменше з l лінійних заборон вхідної системи лінійних заборон $Ax \neq a_0$.

Вихід. «Так», якщо розв'язок існує, «Ні», інакше.

Покажемо, що ця задача належить класу складності $\mathcal{NP}$.

Твердження 3.4. *Задача Max-SLR-decision належить класу задач* $\mathcal{NP}$.

Доведення. Доведення є аналогічним до твердження 3.1.

Існування поліноміального сертифіката. Сертифікатом для цієї задачі як і у випадку SLR-decision є вектор $(x_1, x_2, \ldots, x_n)$.

Поліноміальна перевірка сертифікату. Перевірка сертифікату полягає в підставленні вектору $(x_1, x_2, \ldots, x_n)$ в систему лінійних заборон і підрахунку кількості лінійних заборон, для яких $(x_1, x_2, \ldots, x_n)$ є розв'язком. Розлянемо алгоритм перевірки сертифікату.

Вхід. Вектор $(x_1, x_2, \ldots, x_n)$.

1) Встановити $Counter = 0$.

2) Для кожного $j = \overline{1, m}$:

   а) Перевірити чи виконується $a_1^{(j)}x_1 + a_2^{(j)}x_2 + \ldots + a_n^{(j)}x_n \neq a_0^{(j)}$.

   б) Якщо виконується, то $Counter = Counter + 1$.

3) Якщо $Counter \geq l$, повернути «Так», інакше повернути «Ні».

Вихід. «Так», якщо x є розв'язком щонайменше l лінійних заборон, «Ні» інакше.

Як і у випадку минулої задачі, процедура перевірки потребує $n \cdot m$ операцій множення в полі $\mathbb{F}_{2^k}$, отже є поліноміальною від довжини вхідних даних задачі.

□

Для доведення $\mathcal{NP}$-повноти задачі Max-SLR-decision будемо використовувати задачу Max-XORSAT.

Твердження 3.5. *Задача Max-SLR-decision є $\mathcal{NP}$-складною.*

Доведення. Для доведення необхідно показати, що Max-XORSAT $\leq_p$ Max-SLR-decision.

Нагадаємо яким чином формулюється задача Max-XORSAT. На вхід задачі подається формула, представлена у КНФ, в якій усі операції диз'юнкції в диз'юнктах замінені на операцію XOR. Формула містить n змінних $(x_1, x_2, \ldots, x_n)$ та m виразів зі XOR, які для скорочення будемо називати XOR-виразами. В XOR-вираз може входити як змінна, так і її заперечення. Також на вхід подається l, де l — натуральне число, яке не перебільшує кількості XOR-виразів. Необхідно визначити чи існує набір значень, який задовольняє щонайменше l XOR-виразів в заданій формулі.

Цю формулу можна записати у вигляді системи рівнянь, прирівнявши кожний XOR-вираз до одиниці, тоді питання буде полягати в тому, чи можна знайти набір, який є розв'язком для щонайменше l рівнянь цієї системи. Слід зауважити, що усі операції заперечення в XOR-виразах можна замінити на XOR змінної з одиницею. Тоді в лівих частинах рівнянь залишаться лише змінні, а праві частини рівнянь будуть дорівнювати нулю у випадку, коли в XOR-виразі містилась непарна кількість змінних з запереченням, та одиниці у всіх інших випадках.

Отже, система буде мати такий вигляд:

$$\begin{cases} x_{i_1^{(1)}} + x_{i_2^{(1)}} + \ldots + x_{i_{k_1}^{(1)}} = y_1 \\ x_{i_1^{(2)}} + x_{i_2^{(2)}} + \ldots + x_{i_{k_2}^{(2)}} = y_2 \\ \ldots \\ x_{i_1^{(m)}} + x_{i_2^{(m)}} + \ldots + x_{i_{k_m}^{(m)}} = y_m \end{cases},$$

де $1 \leq k_l \leq n$ для $l = \overline{1,m}$ — це кількість змінних в кожному з XOR-виразів, $i_j^{(l)} \in \{1, 2, \ldots, n\}$ для $1 \leq j \leq k_l$, $l = \overline{1,m}$ — це індекси, які задають змінні, що потрапили в кожний з XOR-виразів (зауважимо, що в XOR-виразі не можуть міститись змінні з однаковими індексами або змінна та її заперечення одночасно), $y_l \in \{0, 1\}$ для $l = \overline{1,m}$ — це набір значень, які приймають відповідні XOR-вирази.

Бачимо, що індекси в такому записі фактично відображають присутність деякої змінної з набору $(x_1, x_2, \ldots, x_n)$ в деякому з m рівнянь. Цей запис можна спростити, якщо ввести штучні змінні, кожна з яких буде індикатором того, чи присутня деяка змінна в деякому рівнянні. Тоді ця система набуває такого виду:

$$\begin{cases} a_1^{(1)} x_1 + a_2^{(1)} x_2 + \ldots + a_n^{(1)} x_n = y_1 \\ a_1^{(2)} x_1 + a_2^{(2)} x_2 + \ldots + a_n^{(2)} x_n = y_2 \\ \ldots \\ a_1^{(m)} x_1 + a_2^{(m)} x_2 + \ldots + a_n^{(m)} x_n = y_m \end{cases},$$

де $a_i^{(j)} \in \{0, 1\}$ для $i = \overline{1,n}$, $j = \overline{1,m}$ — ці штучні змінні та $y_j \in \{0, 1\}$ для $j = \overline{1,m}$. Зауважимо, що таке перетворення впливає лише на представлення системи рівнянь та не змінює множину розв'язків.

Оскільки y_i для $i = \overline{1,m}$ приймає лише два значення, то можемо

позначити $\widetilde{y}_i = y_i + 1$ для $i = \overline{1, m}$. Тоді система має вигляд:

$$\begin{cases} a_1^{(1)}x_1 + a_2^{(1)}x_2 + \ldots + a_n^{(1)}x_n \neq \widetilde{y}_1 \\ a_1^{(2)}x_1 + a_2^{(2)}x_2 + \ldots + a_n^{(2)}x_n \neq \widetilde{y}_2 \\ \ldots \\ a_1^{(m)}x_1 + a_2^{(m)}x_2 + \ldots + a_n^{(m)}x_n \neq \widetilde{y}_m \end{cases}.$$

Отримали систему лінійних заборон над полем $\mathbb{F}_2$. Отже, задача Max-XORSAT фактично є частковим випадком задачі Max-SLR-decision. Функція зведення f, яка ставить у відповідність кожному екземпляру задачі Max-XORSAT екземпляр задачі Max-SLR-decision, — це описане вище перетворення булевої формули з XOR-виразами у систему лінійних заборон над $\mathbb{F}_2$.

Функція f зберігає при відображенні множину екземплярів задачі Max-XORSAT з відповіддю «Так», тобто Max-XORSAT$(x) = 1$ тоді й тільки тоді, коли Max-SLR$(x) = 1$. Це випливає з побудови функції f — на кожному кроці виконувались лише еквівалентні перетворення початкової булевої формули, тому в результаті була побудована система лінійних заборон, в якої множина розв'язків співпадає з множиною входів булевої формули, на яких вона приймає значення 1. Складність обчислення функції f обмежена поліномом від вхідних даних, оскільки вона потребує лише декількох ітерацій матриці A та вектору a_0. □

Наслідок 3.1. *Задача Max-SLR-decision є $\mathcal{NP}$-повною.*

Сформулюємо версію задачі SLR-decision, наклавши обмеження $m \leq 2^{k-1}$ на кількість лінійних заборон в системі. Зауважимо, що сформульована таким чином задача є *задачею типу promise*. Це означає, що в рамках цієї задачі не на всі можливі вхідні дані необхідно надавати відповідь «Так» або «Ні» — на частину входів, для яких $m > 2^{k-1}$, можна нічого не повертати.

Задача 3.5 (Restricted-SLR-decision)**.** *Вхід.* $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$ над полем $\mathbb{F}_{2^k}$, вектор a_0 розміру $m \times 1$, де $m \leq 2^{k-1}$.

Необхідно з'ясувати чи існує розв'язок над полем $\mathbb{F}_{2^k}$ у системи лінійних заборон $Ax \neq a_0$.

Вихід. «Так», якщо розв'язок існує, «Ні», інакше.

Покажемо, що задача Restricted-SLR-decision належить класу складності $\mathcal{RP}$.

Твердження 3.6. *Задача Restricted-SLR-decision належить класу задач* $\mathcal{RP}$.

Доведення. Щоб довести цей факт, необхідно показати, що існує поліноміальний імовірнісний алгоритм B, такий що:

1) якщо Restricted-SLR$(\mathbb{F}_{2^k}, A, a_0) = 1$, то $Pr[B(\mathbb{F}_{2^k}, A, a_0) = 1] \geq \frac{1}{2}$;

2) якщо Restricted-SLR$(\mathbb{F}_{2^k}, A, a_0) = 0$, то $Pr[B(\mathbb{F}_{2^k}, A, a_0) = 1] = 0$.

Побудуємо такий алгоритм B.

Вхід. $\mathbb{F}_{2^k}$, матриця A розміру $m \times n$, вектор a_0 розміру $m \times 1$, де $m \leq 2^{k-1}$.

1) Випадково рівноймовірно генеруємо $(r_1, r_2, \ldots, r_n)$, де $r_i \in \mathbb{F}_{2^k}$ для $i = \overline{1, n}$.

2) Обчислюємо $F(r_1, r_2, \ldots, r_n)$, де $F(x) = \prod_{i=1}^{m} [(a^{(i)}, x) + a_0^{(i)}]$.

3) Якщо $F(r_1, r_2, \ldots, r_n) \neq 0$, то повертаємо «Так», інакше повертаємо «Ні».

Вихід. «Так», якщо існує розв'язок системи $Ax \neq a_0$, «Ні», інакше.

В цьому випадку скористались критерієм існування розв'язків системи лінійних заборон 2.1: розв'язок існує тоді й тільки тоді, коли поліном, що складається з добутку усіх лівих частин, тотожно не дорівнює нулю.

Цей алгоритм є поліноміальним від довжини входу, оскільки потребує $m \times n$ операцій множення в полі.

Переконаємось у виконанні умов.

1) Припустимо, що Restricted-SLR$(\mathbb{F}_{2^k}, A, a_0) = 0$, тоді $F(x) \equiv 0$. В такому разі у всіх можливих випадках буде відповідь «Ні», оскільки для полінома, тотожно рівного нулю, не існує набору значень, при якому його

значення буде відрізнятись від нуля.

2) Припустимо, що Restricted-SLR$(\mathbb{F}_{2^k}, A, a_0) = 1$, тоді $F(x) \not\equiv 0$. В такому разі імовірність того, що $F(x)$ буде дорівнювати нулю можна оцінити за допомогою леми Шварца-Зіпеля:

$$Pr_{r_1,r_2,\ldots,r_n \in \mathbb{F}_{2^k}}[F(r_1, r_2, \ldots, r_n) = 0] \leq \frac{m}{2^k} \leq \frac{1}{2}.$$

Можемо обчислити імовірність протилежної події:

$$Pr_{r_1,r_2,\ldots,r_n \in \mathbb{F}_{2^k}}[F(r_1, r_2, \ldots, r_n) \neq 0] \geq \frac{1}{2}.$$

Оскільки відповідь «Так» надається лише у випадку, коли поліном не дорівнює нулю, то імовірність того, що алгоритм B надасть відповідь «Так», більша за 0.5.

Матриця помилок алгоритму наведена в таблиці 3.1.

Таблиця 3.1 – Матриця помилок алгоритму розв'язку задачі Restricted-SLR-decision

	«Так»	«Ні»
Restricted-SLR$(y) = 1$	$\geq \frac{1}{2}$	$\leq \frac{1}{2}$
Restricted-SLR$(y) = 0$	0	1

Обчислення помилок алгоритму завершує доведення.

□

З огляду на розглянуті часткові випадки SLR-decision та аналітичні властивості систем лінійних заборон, виникло дві гіпотези щодо уточнення складності задачі SLR-decision.

Перша гіпотеза полягає в тому, що задача SLR-decision є $\mathcal{NP}$-повною. На користь цієї гіпотези можна навести такі спостереження.

1) Множина розв'язків системи лінійних заборон не має певної

структури, як у випадку систем лінійних рівнянь. Це вказує на те, що перевірка існування розв'язку довільної системи лінійних заборон в загальному випадку потребує експоненційного обчислення.

2) Систему лінійних заборон можна представити у вигляді системи квадратичних рівнянь над скінченним полем, а задача MQ в загальному випадку є $\mathcal{NP}$-повною. Звісно, при перетворенні системи лінійних заборон на систему квадратичних рівнянь отримана система має особливу структуру, але наразі не відомо як скористатись цією структурою.

3) У випадку $\mathcal{NP}$-повної задачі SAT її спрощена версія XORSAT належить класу $\mathcal{P}$, а її апроксимаційна спрощена версія Max-XORSAT є $\mathcal{NP}$-повною задачею. Така сама ситуація з частковими випадками $\mathbb{F}_2$-SLR-decision та Max-SLR-decision. Звісно, у задачі Max-SLR-decision вибір поля не обмежується полем $\mathbb{F}_2$, але при доведенні $\mathcal{NP}$-повноти достатньо лише випадку $\mathbb{F}_2$.

Друга гіпотеза полягає в тому, що для задачі SLR-decision існує поліноміальний, можливо імовірнісний, алгоритм розв'язку. На користь цієї гіпотези можна навести такі спостереження.

1) Всі ліві частини заборон в системі є лінійними відносно XOR. Тому не можна застосувати міркування, аналогічні до доведення $\mathcal{NP}$-повноти задачі MQ, оскільки в задачі 3SAT кожний диз'юнкт в КНФ має нелінійну відносно XOR структуру.

2) Задача SLR-decision еквівалентна задачі перевірки тотожної рівності полінома нулю у випадку скінченного поля, а та ж сама задача у випадку нескінченного поля належить класу складності $\mathcal{RP}$.

3) Задача Restricted-SLR-decision належить класу складності $\mathcal{RP}$.

Уточнення оцінки складності для задачі SLR-decision планується у подальших дослідженнях.

Висновки до розділу 3

В даному розділі визначені задачі перевірки існування та пошуку розв'язку системи лінійних заборон над скінченним полем. Доведено еквівалентність за Тюрінгом цих задач, тобто розв'язок однієї з цих задач можна застосувати для розв'язку іншої. Отже, якщо буде побудовано ефективний алгоритм розв'язку SLR-decision, то для задачі SLR-search такий ефективний алгоритм також можна побудувати.

Оскільки доведення еквівалентності задач SLR-decision та SLR-search є конструктивним, то імовірнісний алгоритм перевірки існування розв'язку у випадку $m \leq 2^{k-1}$ було використано для побудови алгоритму пошуку розв'язку у випадку $m \leq 2^{k-1}$. Проведено аналіз цього алгоритму і встановлено, що він є поліноміальним. Також цей алгоритм було імплементовано та перевірено на деякому наборі вхідних даних. На основі алгоритму пошуку розв'язку було побудовано евристичний алгоритм пошуку декількох розв'язків. Цей алгоритм використовує процедуру пошуку ортогонального вектора, щоб виключати з множини розв'язків системи знайдений розв'язок. В загальному випадку, такий алгоритм не є поліноміальним та не гарантує знаходження усіх розв'язків. Цей алгоритм було імплементовано та перевірено на деякому наборі вхідних даних. Також експериментально встановлено, що у випадку достатньої кількості заборон в системі та невеликої потужності множини розв'язків, цей алгоритм відновлює більшість цих розв'язків.

Сформульовано часткові випадки задачі SLR-decision та визначено клас складності для кожного з цих випадків. Запропоновано декілька гіпотез щодо уточнення складності задачі SLR-decision та розглянуті фактори, які потенційно вказують на кожну з цих гіпотез. Розглянуті фактори враховують структуру систем лінійних заборон, наявні алгебраїчні властивості систем лінійних заборон та часткові випадки, для яких визначено класи складності.

ВИСНОВКИ

В ході дослідження виконано такі завдання.

1) Формалізовано задачу відновлення невідомого вектора за частковою інформацією, представленою у формі лінійних залежностей, за допомогою введення нотації системи лінійних заборон над скінченним двійковим полем. Сформульовано та доведено критерій існування розв'язку системи лінійних заборон. На основі цього критерію побудовано поліноміальний імовірнісний алгоритм перевірки існування розв'язку для випадку $m \leq 2^{k-1}$. Проведено аналіз цього алгоритму та обчислено імовірність помилки, яка є односторонньою і становить 2^{-d}, де d — це фіксоване число, яке визначає порядок точності алгоритму. Цей алгоритм було імплементовано та перевірено на певних наборах вхідних параметрів.

2) Розглянуто випадок, коли систему лінійних заборон згенеровано невідомим фіксованим вектором, та досліджено принципову можливість відновлення невідомого вектора в такому випадку. Це дослідження надало змогу побудувати систему лінійних заборон, яка містить невелику кількість лінійних заборон (у порівнянні з кількістю усіх можливих лінійних заборон в системі), але має єдиний розв'язок. Цей результат надав змогу отримати нетривіальну оцінку на точку насичення у випадку ненульових правих частин.

3) Сформульовано задачі перевірки існування розв'язку та пошуку розв'язку для системи лінійних заборон. Доведено еквівалентність цих задач за Тюрінгом, що надало змогу побудувати поліноміальний імовірнісний алгоритм пошуку хоча б одного розв'язку системи лінійних заборон у випадку $m \leq 2^{k-1}$. Алгоритм було імплементовано та апробовано на певних наборах вхідних даних. На основі цього алгоритму було побудовано інший евристичний алгоритм для пошуку декількох розв'язків системи лінійних заборон. В ході апробації було виявлено, що найкращі результати цей алгоритм показує у випадку, коли потужність

множини розв'язку системи лінійних заборон становить приблизно $1 - 10$ векторів.

4) Встановлено клас складності задачі перевірки існування розв'язку системи лінійних заборон. Також для цієї задачі сформульовано певний набір задач, які є частковими випадками початкової задачі. Для всіх часткових випадків встановлено їх приналежність відповідним класам складності. Ці результати надали змогу висунути гіпотези щодо уточнення складності задачі перевірки існування розв'язку системи лінійних заборон.

В подальших дослідженнях планується виконати пошук поліноміального імовірнісного алгоритму перевірки існування розв'язку системи лінійних заборон над скінченним полем в загальному випадку. Також планується дослідити можливість уточнення оцінки складності задачі перевірки існування розв'язку системи лінійних заборон.

ПЕРЕЛІК ПОСИЛАНЬ

1. Menezes A. Handbook of Applied Cryptography / A. Menezes, S. Vanstone, P. Oorschot. – Boca Raton: CRC Press, Inc., 1996. – 816 с. – (Discrete Mathematics and Its Applications).

2. Bernstein D. The Salsa20 Family of Stream Ciphers / Daniel Bernstein // New Stream Cipher Designs: The ESTREAM Finalists / Daniel Bernstein. – Berlin: Springer-Verlag, 2008. – С. 84–97.

3. Cannière C. Trivium / C. Cannière, B. Preneel // New Stream Cipher Designs: The ESTREAM Finalists / C. Cannière, B. Preneel. – Berlin: Springer-Verlag, 2008. – С. 244–266.

4. Lo C. Short Communication: Stream Ciphers for GSM Networks / C. Lo, Y. Chen. // Comput. Commun.. – 2001. – №11. – С. 1090–1096.

5. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C / B. Schneier, P. Sutherland. – New York: John Wiley & Sons, Inc., 1995. – 758 с.

6. Cox D. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra / D. Cox, J. Little, D. O'Shea. – Berlin: Springer-Verlag, 2007. – 553 с. – (Undergraduate Texts in Mathematics).

7. Ars G. Algebraic Immunities of functions over finite fields / G. Ars, J. Faugère. – 2005.

8. Dubé T. The Structure of Polynomial Ideals and Gröbner Bases / Thomas Dub́e. // Society for Industrial and Applied Mathematics. – 1990. – №19. – С. 750–773.

9. Faugère J. A new efficient algorithm for computing Gröbner bases (F4) / Jean-Charles Faugère. – 1999.

10. Faugère J. A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5) / Jean Charles Faugère // Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation / Jean

Charles Faugère. – New York: Association for Computing Machinery, 2002. – (ISSAC; 2). – С. 75–83.

11. Faugére J. On the Complexity of the F5 Gröbner basis Algorithm [Електронний ресурс] / J. Faugére, M. Bardet, B. Salvy. – 2013. – Режим доступу до ресурсу: https://arxiv.org/abs/1312.1655.

12. Bard G. Algebraic Cryptanalysis / Gregory Bard., 2009. – (392).

13. Courtois N. About the XL Algorithm over GF(2) / N. Courtois, J. Patarin // Proceedings of the 2003 RSA Conference on The Cryptographers' Track / N. Courtois, J. Patarin. – San Francisco: Springer-Verlag. – (CT-RSA; 3). – С. 141–157.

14. ElimLin Algorithm Revisited / N.Courtois, P. Sepehrdad, P. Sušil, S. Vaudenay // Proceedings of the 19th International Conference on Fast Software Encryption / N.Courtois, P. Sepehrdad, P. Sušil, S. Vaudenay. – Washington: Springer-Verlag, 2012. – (FSE; 12). – С. 306–325.

15. Råddum H. New Technique for Solving Sparse Equation Systems [Електронний ресурс] / H. Råddum, I. Semaev. – 2006. – Режим доступу до ресурсу: https://eprint.iacr.org/2006/475.pdf.

16. Ding J. Zhuang-Zi: A New Algorithm for Solving Multivariate Polynomial Equations over a Finite Field [Електронний ресурс] / J. Ding, J. Gower, D. Schmidt. – 2006. – Режим доступу до ресурсу: https://eprint.iacr.org/2006/038.pdf.

17. Bard G. Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over GF(2) via SAT-Solvers [Електронний ресурс] / G. Bard, N. Courtois, C. Jefferson. – 2001. – Режим доступу до ресурсу: https://eprint.iacr.org/2007/024.pdf.

18. McDonald C. An Algebraic Analysis of Trivium Ciphers based on the Boolean Satisfiability Problem [Електронний ресурс] / C. McDonald, C. Charnes, J. Pieprzyk. – 2007. – Режим доступу до ресурсу: https://eprint.iacr.org/2007/129.pdf.

19. A short overview on modern parallel SAT-solvers [Електронний ресурс] / [S. Hölldobler, N. Manthey, V. Nguyen та ін.]. – 2011. – Режим

доступу до ресурсу: https://www.researchgate.net/publication/254048189_A_short_overview_on_modern_parallel_SAT-solvers.

20. Garey M. Computers and Intractability: A Guide to the Theory of NP-Completeness / M. Garey, D. Johnson. – New York: W. H. Freeman & Co., 1979. – 338 с.

21. Cormen T. H. Introduction to Algorithms / T. H. Cormen, C. E. Leiserson, R. L. Rivest., 2009. – 1320 с.

22. Arora S. Computational Complexity: A Modern Approach / S. Arora, B. Barak. – New York: Cambridge University Press, 2009. – 608 с.

23. Cook S. The Complexity of Theorem-Proving Procedures / Stephen Cook. // Proceedings of the Third Annual ACM Symposium on Theory of Computing. – 1971. – №71. – С. 151–158.

24. Karp R. Reducibility among combinatorial problems / Richard Karp // Complexity of Computer Computations / Richard Karp. – New York: Plenum Press, 1972. – С. 85–103.

25. Moore C. The Nature of Computation / C. Moore, S. Mertens. – New York: Oxford University Press, Inc., 2011. – 1032 с.

26. Fraenkel A. Complexity of problems in games, graphs and algebraic equations / A. Fraenkel, Y. Yesha. // Discrete Applied Mathematics. – 1979. – №1. – С. 15–30.

27. Fraenkel A. Complexity of Solving Algebraic Equations / A. Fraenkel, Y. Yesha. // Inf. Process. Lett.. – 1980. – №10. – С. 178–179.

28. Simple Matrix - A Multivariate Public Key Cryptosystem (MPKC) for Encryption / C.Tao, H. Xiang, A. Petzoldt, A. Ding. // Finite Fields Appl.. – 2015. – №35. – С. 352–368.

29. Kipnis A. Unbalanced Oil and Vinegar Signature Schemes / A. Kipnis, J. Patarin, L. Goubin. // Advances in Cryptology — EUROCRYPT '99. – 1999. – С. 206—222.

30. Ding J. Rainbow, a New Multivariable Polynomial Signature Scheme / J. Ding, D. Schmidt // Proceedings of the Third International Conference on Applied Cryptography and Network Security / J. Ding, D. Schmidt. – New

York: Springer-Verlag, 2005. – (ACNS; 5). – С. 164–175.

31. Solving Underdefined Systems of Multivariate Quadratic Equations [Електронний ресурс] / N.Courtosis, L. Goubin, W. Meier, J. Tacier. – 2002. – Режим доступу до ресурсу: http://www.goubin.fr/papers/CG02.pdf.

32. Huang H. Algorithm for Solving Massively Underdefined Systems of Multivariate Quadratic Equations over Finite Fields [Електронний ресурс] / H. Huang, W. Bao. – 2015. – Режим доступу до ресурсу: http://arxiv.org/abs/1507.03674.

33. Goldreich O. Computational Complexity: A Conceptual Perspective / Oded Goldreich. – New York: Cambridge University Press, 2008. – 632 с.

34. Вінберг Е. Б. Курс алгебри / Ернест Борисович Вінберг. – Москва: МЦНМО, 2019. – 592 с.

35. Lewin D. Checking Polynomial Identities over any Field: Towards a Derandomization? / D. Lewin, S. Vadhan // Proceedings of 30th Annual ACM Symposium on Theory of Computing / D. Lewin, S. Vadhan. – Dallas: ACM, 1998. – С. 438–447.